CapDEM TD - Systems Engineering

State of the Art Report

Attilio Bagnulo
CGI

Jocelyn Leclerc
CGI

Scient. Autho.: Mario Couture
Tél.: (418) 844-400 (4285)

Defence Research and Development Canada, Valcartier

Contract Report W7701-3-2621

DRDC – Valcartier CR 2004-428

2005-03-03

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2005**	2. REPORT TYPE	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **CapDEM TD -Systems Engineering State of the Art Report**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defence R&D Canada -Ottawa,National Defence Headquarters,Ottawa,CA,K1A OK2**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**73**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Approved by

Mario Couture, DRDC-Valcartier

Defence Scientist in System Engineering

Abstract

This report provides a state-of-the-art review of Systems Engineering standards and approaches as they may apply to the complex nature of military Systems of Systems. It contains an analysis of current practices and trends regarding standards and maturity models, available handbooks and guides, and the most widely used approaches. The report also discusses generic problems and issues encountered in the world of military systems including particular problems and issues that specifically apply to DND/CF context. Finally, the report provides a set of recommendations to be applied within the context of CapDEM TD in light of the knowledge acquired during the state-of-the-art review.

Résumé

Ce rapport fournit une revue de l'état de l'art concernant les différentes normes et approches d'ingénierie de systèmes pouvant s'appliquer à la nature complexes des systèmes de systèmes militaires. Il contient une analyse des pratiques et tendances actuelles concernant les standards et modèles de maturité, un survol des manuels et guides ainsi qu'une description des approches les plus utilisées. Le rapport discute également des problèmes et enjeux génériques associés aux systèmes militaires incluant un rappel des problèmes et enjeux qui sont spécifiques au contexte MDN/FC. Finalement le rapport énonce un ensemble de recommandations pouvant s'appliquer au contexte de DIGCap DT, à la lumière des connaissances acquises au cours de la revue de l'état de l'art.

This page intentionally left blank.

Executive Summary

The Collaborative Capability Definition Engineering and Management Technology Demonstrator Program (CapDEM TD) initiative is investigating Capability Engineering (CE) in order to improve an acquisition process (15+ years) that does not meet the rapid evolution of business requirements and technology, that does not support the new capability-based approach promoted by DND/CF and that is not adapted for the increasing complexity of Systems of Systems. To that end, a new Capability Engineering Process (CEP) is required. The CEP will be based on:

- Best practices and standards issued from traditional Systems Engineering (SysEng);
- Best practices in Systems of Systems (SoS);
- Best practices in Simulation-based acquisition (SBA); and
- Tools and technologies facilitating data/information exchange and collaboration among engineers, scientists, users and managers at multiple distributed geographic-locations for the purpose of defining, develop and evaluate a capability.

State-of-the-art reviews with respect to SysEng, Modeling & Simulation (M&S), Tools and Technologies, and, Systems of Systems Engineering (SoSEng) will provide input to the definition of the CEP.

The purpose of this document is to perform a review of the current state-of-the-art of SysEng practices[1]. This document describes SysEng concepts, identifies the current enabling standards, processes, methods, frameworks and approaches used in engineering a system. In addition, a discussion is presented on the challenges in the acquisition of defence capabilities. Recommendations are identified for the CapDEM TD initiative on the potential SysEng practices to be developed and implemented that can address these challenges within a SoS environment.

Systems Engineering Defined

The document describes the contemporary understanding of SysEng and some of the major concepts including engineering, systems, process, and system life cycle. SysEng is an interdisciplinary collaborative approach to derive, evolve and verify the life cycle balanced system solution to meet customer expectations and public acceptability. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem:

- Operations;
- Performance;
- Test;
- Manufacturing;

[1] The report presents the results of a 40-days effort summarizing the state of the art of the SysEng field. It is in no way exhaustive and reflects the most obvious findings and conclusions that could be obtained in this short period of time.

- Cost & Schedule;
- Training & Support; and
- Disposal.

Current Practices and Trends

Several standards embody the current practices in SysEng. The current SysEng Standards are the ISO/IEC 15288 Systems Engineering – System Life Cycle Processes [3], EIA 632 Processes for Engineering a System [6], and IEEE 1220-1998 Standard for Application and Management of the Systems Engineering Process [2]. The ISO/IEC 15288 addresses the high level SysEng processes through the complete life cycle stages of a product or service being developed from conception to retirement. The EIA 632 provides the next level of detail mainly for the concept, development and production life cycle stages. The IEEE 1220 provides the next level of detail of the SysEng process and specifically covers the development life cycle stage but does not cover the breadth of the life cycle as addressed in the other two standards. The Capability Maturity Model (CMMI) [14] provides the framework to measure the effectiveness of the SysEng process of a particular organization or unit. In addition, Domain-Specific Standards and Frameworks are required to provide the next level of detail, and are applied according to the nature of the system element to be developed.

These standards do not specify the details of "how to" implement process requirements for engineering a system. In the same manner, they do not mandate any specific method or tool that a developer could use to implement the specific needs of a process. Several handbooks and guides are identified here that do describe the processes to be applied for a particular industry: ISO/IEC 19760 Systems Engineering – A guide for the application of ISO/IEC 15288 [10], NASA Systems Engineering Handbook [11], UK MoD Defence Systems Engineering Handbook [23], UK MoD Acquisition Management System [26], INCOSE Systems Engineering Handbook [28], Software Productivity Consortium's Integrated Systems and Software Engineering Process (ISSEP) [12], and the US DoD Systems Engineering Fundamentals [41].

In addition to these standards, there are certain approaches that identify best practices in the application of these SysEng processes. The approaches discussed in this report are:

- Sequential;
- Incremental;
- Evolutionary;
- Agile;
- Architecture-driven;
- Modelling and simulation;
- Concurrent engineering;
- Spiral development;
- Open systems; and
- Product improvement.

DND/CF Challenges

The challenges described in this report are either of a generic nature, applying to DND/CF as well as to most military organizations (e.g. US, UK, Australia), or specific to DND/CF as previously reported in our analysis of the current situation [40].

The generic challenges are:

- Military vs. Commercial Context;
- Capability vs. System Focus;

- Systems of Systems vs. Single System;
- Long Acquisition Cycle;
- Changing Requirements; and
- Integrated and Dispersed Project Teams.

Specific challenges are:

- Current equipment Platform-centric approach to Strategic Planning (rather than capability-based);
- Current approach to Requirements Definition (e.g. "bottom-up", absence of coherent overall plan, inefficient process);
- Current approach to Capability Acquisition (e.g. "one-size fits all" approach, risk aversion, slow and cumbersome process, responsibility overlap, etc.); and
- The lack of formal methods (and clear timelines).

Recommendations for CapDEM TD

In order to address the many challenges in acquiring systems within the Defence community, CapDEM TD should further explore the following recommendations:

- Apply specific SysEng principles at every stage of the acquisition cycle and to the supporting management and technical processes;
- Implement a CEP within an overarching Capability Management Framework;
- Implement an overall systems integration framework that addresses SysEng Organization, Infrastructure, Management, Interoperability and Architecture;
- Adopt an Evolutionary Systems Engineering Approach that first delivers an initial operational capability (IOC) followed by incremental capabilities that bring successive refinements until the final operational capability (FOC) is achieved as planned;
- Maintain a link between the changing users needs and the actual capability achieved throughout the entire system life cycle;
- Adopt and enforce the concepts of concurrent engineering;
- Define clear organizational responsibilities to warrant the good stewardship of the above environment and procedures;
- Define appropriate standards-based processes and procedures to guide the whole spectrum of through life SysEng activities;
- Acquire and use a repository-based collaborative engineering environment;
- Conduct adequate Change Management and Training programs to ensure the adoption, appropriate use, and continuous evolution of the new environment;
- Extend and/or adapt the current SysEng standards and approaches to develop a Canadian CEP placing the emphasis on capabilities and providing support to a top-down and through life approach to requirements definition and refinement; and
- Implement the Capability Maturity Model Integrated (CMMI) for Systems and Software Engineering.

Sommaire

L'initiative d'ingénierie collaborative et de gestion des capacités (DIGCap) investigue la possibilité de faire appel à l'ingénierie des capacités (IngCap) pour améliorer un processus d'acquisition (15ans +) qui ne rencontre plus l'évolution rapide des besoins d'affaires et de la technologie, qui ne supporte pas la nouvelle approche orientée-capacités mise d'avant par le Ministère de la défense nationale (MDN) et les forces canadiennes (FC), et finalement, qui n'est pas adapté à la complexité croissante des systèmes de systèmes (SdS). À cette fin, un nouveau processus d'ingénierie des capacités (PIC) est requis. Le futur PIC sera basé sur :

- Les meilleures pratiques et normes émanant de l'ingénierie de systèmes traditionnelle (IngSys);
- Les meilleures pratiques en matière d'ingénierie de SdS;
- Les meilleures pratiques en matière d'acquisition basée sur la simulation (ABS); et
- Les outils et technologies facilitant les échanges d'information ainsi que la collaboration entre les différents ingénieurs, scientifiques, utilisateurs et gestionnaires répartis géographiquement, dans le but de définir, développer et évaluer les capacités.

Les intrants nécessaires à la définition du PIC seront fournis par les résultats des revues de l'état de l'art portant sur l'IngSys, la modélisation et simulation (M&S), les outils et la technologie d'ingénierie collaborative, ainsi que les SdS.

Le but de ce document est de produire une revue de l'état de l'art sur les pratiques d'IngSys[2]. Le document décrit les concepts d'IngSys, identifie les normes, processus, méthodes et cadres conceptuels sous-jacents, ainsi que les approches utilisées pour développer un système. On y discute également les enjeux que présente l'acquisition de nouvelles capacités dans le contexte de la défense. Des recommandations sont identifiées à l'intention de l'initiative DIGCap DT au regard des pratiques potentielles d'IngSys à être développées et implantées dans le but de résoudre les enjeux inhérents aux environnements de SdS.

Définition de l'ingénierie de systèmes

Le document décrit la compréhension contemporaine de l'IngSys ainsi que quelques concepts majeurs incluant l'ingénierie, les systèmes, les processus et le cycle de vie. L'IngSys est une approche collaborative interdisciplinaire visant à déduire, évoluer et vérifier une solution système équilibrée quant à son cycle de vie dans le but de satisfaire les attentes du client et l'assentiment du public. Elle place l'emphase sur la définition et la documentation des besoins du client et des fonctions requises tôt dans le cycle de développement, puis sur la synthèse des besoins et la validation tout en considérant l'ensemble de la problématique :

[2] Le rapport présente les résultats de 40-jours d'effort résummant l'état de l'art dans le domaine de l'IngSys. Il n'est d'aucune manière exaustif et reflète les constatations et conclusions les plus évidentes qui purent être obtenues dans cette courte période de temps.

- Opérations;
- Performance;
- Essais;
- Fabrication;
- Coûts et échéances;
- Formation et support, et
- Fin de vie.

Pratiques et tendances actuelles

Plusieurs normes encadrent les pratiques courantes en matière d'IngSys. Ces normes sont: "ISO/IEC 15288 Systems Engineering – System Life Cycle Processes" [3], "EIA 632 Processes for Engineering a System" [6] et "IEEE 1220-1998Standard for Application and Management of the System Engineering Process" [2]. La norme ISO/IEC 15288 concerne les processus d'ingénierie de systèmes de haut niveau et couvre toutes les étapes du cycle de vie complet de développement d'un produit ou d'un service de sa conception à sa fin de vie. La norme EIA 632 couvre le niveau de détail suivant principalement pour les étapes du cycle de vie qui concernent la conception, le développement et la production. La norme IEEE 1220 traite du niveau de détail suivant du processus d'IngSys et concerne plus spécifiquement l'étape de développement du cycle de vie mais n'a pas la même portée que les deux autres normes quant à sa couverture du cycle de vie. Les modèles de maturité des capacités (CMMI) [14] fournissent un cadre de référence pour mesurer l'efficacité du processus d'ingénierie de systèmes d'une organisation particulière ou d'une de ses composantes. Outre ces normes de base, d'autres normes et cadres de référence spécifiques à certains domaines sont requis pour fournir le niveau de détail subséquent. Ces normes et cadres sont appliqués selon la nature de l'élément de système qui est en cours de développement.

Ces normes ne fournissent pas de détails sur la manière d'implanter les besoins particuliers d'un processus d'IngSys. De la même manière, elles ne désignent aucune méthode ou outils qu'un développeur pourrait utiliser pour implanter les besoins particuliers du processus. Plusieurs manuels de références et guides identifiés dans le rapport servent à décrire les processus s'appliquant à une industrie en particulier : "ISO/IEC 19760 System Engineering – A guide for the application of ISO/IEC 15288" [10], "NASA Systems Engineering Handbook" [11], "UK MoD Defence Systems Engineering Handbook" [23], "UK MoD Acquisition Management System" [26], "INCOSE Systems Engineering Handbook" 28], "Software Productivity Consortium's Integrated Systems and Software Engineering Process (ISSEP)"[12] et "US DoD Systems Engineering Fundamentals" [41].

En plus des normes, il existe certaines approches qui identifient les meilleures pratiques pour l'application des processus d'IngSys. Les approches discutées dans ce rapport sont :

- Séquentielle;
- Incrémentale;
- Évolutive;
- Agile;
- Orientée-architectures;
- Modélisation et simulation,
- Ingénierie simultanée;
- Développement en « spirale »;
- Systèmes ouverts et
- Amélioration de produit.

Enjeux reliés au contexte MDN/CF

Les enjeux décrits dans ce rapport sont soit de nature générique, s'appliquant tant au contexte MDN/FC qu'au contexte de la plupart des organisations militaires (ex : ÉU, RU, Australie) ou spécifique au contexte MDN/FC tel que rapporté dans notre analyse de la situation actuelle [40].

Les enjeux génériques sont :

- Contexte militaire plutôt que commercial
- Emphase sur les capacités plutôt que sur les systèmes;
- Systèmes de systèmes versus système individuel;
- Long cycle d'acquisition;
- Besoins évolutifs; et
- Équipes de projet intégrées et distribuées géographiquement.

Les enjeux spécifiques sont :

- Approche courante de planification stratégique centrée sur l'équipement (plutôt que sur les capacités);
- Approche courante de définition des besoins (ex: de bas en haut, sans plan d'ensemble, processus inefficace et inefficient);
- Approche courante d'acquisition des capacités (ex : approche uniforme appliquée sans discrimination, aversion au risque, processus lourd et coûteux, chevauchement dans les responsabilités, etc.); et
- Absence de méthodes formelles (et échéanciers précis).

Recommandations pour CapDEM TD

Dans le but de résoudre les nombreux défis que représente l'acquisition de systèmes dans la communauté de la défense, DIGCap DT devra explorer davantage les recommandations suivantes:

- Appliquer des principes spécifiques à l'IngSys à toutes les étapes du cycle d'acquisition incluant les processus de support pour la gestion et les aspects techniques;
- Implanter un PIC à l'intérieur d'un cadre formel de gestion des capacités;
- Implanter un cadre général d'intégration des systèmes couvrant tous les aspects de l'IngSys soit: L'organisation, les infrastructures, la gestion, l'interopérabilité et l'architecture;
- Adopter une approche évolutive d'IngSys qui verra à livrer une capacité opérationnelle initiale (COI) suivie de capacités incrémentales apportant des raffinements successifs jusqu'à l'atteinte de la capacité opérationnelle totale (COT) telle que planifiée;
- Maintenir un lien entre les besoins changeants des utilisateurs et la capacité courante pendant tout le cycle de vie des systèmes;
- Adopter et appliquer les concepts d'ingénierie simultanée;
- Définir des responsabilités organisationnelles claires pour garantir la bonne gestion de l'environnement et des procédures énoncées précédemment;

- Définir des processus et procédures basées sur des normes pour guider l'ensemble des activités pour tout le cycle de vie d'IngSys;
- Faire l'acquisition d'un environnement collaboratif basé sur un répertoire;
- Développer et réaliser des programmes appropriés de gestion du changement et de formation dans le but d'assurer l'adoption, l'utilisation appropriée ainsi que l'évolution continue du nouvel environnement; et
- Apporter des extensions ainsi que des adaptations aux normes et approches courantes d'IngSys afin de développer un PIC pour MND/FC qui placera l'emphase sur les capacités et qui permettra la définition des besoins du haut vers le bas et leur raffinement pendant tout le cycle de vie des systèmes;
- Implanter le modèle 'CMMI' pour l'ingénierie des systèmes et des logiciels [14].

Table of contents

1. Introduction........15

 1.1 Document Objective........15

 1.2 Background........15

 1.3 Systems Engineering Defined........16

  1.3.1 Engineering........16

  1.3.2 Systems........16

  1.3.3 Systems Engineering........17

  1.3.4 Process........19

  1.3.5 System Life Cycle........19

 1.4 Overview........19

2. Current Practices and Trends........22

 2.1 Systems Engineering Standards........23

  2.1.1 ISO/IEC 15288 Systems Engineering – System Life Cycle Processes23

  2.1.2 Electronics Industries Alliance (EIA) 632: Processes for Engineering a System........26

  2.1.3 IEEE 1220-1998 – Applications and Management of the Systems Engineering Process........29

 2.2 Capability Maturity Models........32

 2.3 Domain-Specific Standards and Frameworks........33

 2.4 Systems Engineering Handbooks and Guides........34

  2.4.1 ISO/IEC 19760 Systems Engineering - A Guide for the Application of ISO/IEC 15288 (System Life Cycle Processes)........34

  2.4.2 NASA Systems Engineering Handbook........34

  2.4.3 UK MoD Defence Systems Engineering Handbook........34

  2.4.4 UK MoD Acquisition Management System........35

  2.4.5 INCOSE Systems Engineering Handbook........35

  2.4.6 Integrated Systems and Software Engineering Process (ISSEP)........35

  2.4.7 US DoD Systems Engineering Fundamentals........35

 2.5 Systems Engineering Approaches........36

  2.5.1 Sequential Approach........36

2.5.2 Incremental Approach.....37
2.5.3 Evolutionary Approach.....37
2.5.4 Agile Systems Approach.....38
2.5.5 Architecture-Driven Approach.....40
2.5.6 Modelling and Simulation.....41
2.5.7 Concurrent Engineering.....41
2.5.8 Spiral Development Method.....41
2.5.9 Open Systems Approach.....43
2.5.10 Product Improvement Strategies.....44

3. DND/CF Challenges.....45
3.1 Generic Challenges.....45
3.1.1 Military vs. Commercial Context.....45
3.1.2 Capability vs. Systems Focus.....47
3.1.3 SoS vs. Single System.....49
3.1.4 Long Acquisition Cycle.....53
3.1.5 Changing Requirements.....53
3.1.6 Integrated and Dispersed Project Teams.....53
3.2 Specific Challenges.....54
3.2.1 Equipment Platform-centric Approach to Strategic Planning.....54
3.2.2 Approach to Requirements Definition.....54
3.2.3 Approach to Capability Acquisition.....55
3.2.4 Formal Methods.....55

4. Recommendations for CapDEM TD.....56
4.1 Address the Specific Nature of Military Systems.....56
4.2 Focus on Capabilities.....56
4.3 Address SoS Challenges.....56
4.4 Adopt Evolutionary Systems Engineering.....57
4.5 Adopt Collaborative Engineering.....58
4.6 Review Current Approaches.....58
4.7 Adopt Formal Methods.....58

5. Conclusion.....60

6. References 61

List of figures

Figure 1: Capability-based planning - today [44] 15

Figure 2: Capability-Based Planning - the proposed solution [44] 16

Figure 3: Hierarchy of elements within a system Systems Engineering [3] 17

Figure 4: The systems engineering environment within an enterprise [3] 18

Figure 5: Enterprise and engineering views related to system life cycle stages [10] 20

Figure 6: Relationship of Systems Engineering standards to life cycle stages [3] 21

Figure 7: The frameworks quagmire . [45] 23

Figure 8: ISO/IEC 15288 Systems engineering system life cycle processes [3]. 24

Figure 9: EIA 632 Relationship of processes for engineering a system [6] 27

Figure 10: IEEE 1220-1998 System Engineering Process [2]. 30

Figure 11: Using the DoD Architecture Framework in Systems of Systems Engineering and Acquisition [52, Slide 4] 40

Figure 12: Barry Boehm's Original Spiral Development Model [48] 42

Figure 13: CapDEM TD Definition of Capability Engineering [49, Slide 17] 47

Figure 14: System Complexity Continuum [50, Slide 20] 48

Figure 15: Impact of Capability Engineering on the Systems Engineering Cycle [49, Slide 21]49

Figure 16: LISI levels of interoperability sophistication [42] 52

List of tables

Table 1: Mapping of ISO 15288 system life cycle process groups to the EIA 632 and IEEE 1220 engineering processes. 21

Table 2: Systems Engineering Subprocesses 30

Table 3: Sample domain-specific standards and frameworks 33

This page intentionally left blank.

1. Introduction

1.1 Document Objective

The purpose of this document is to perform a review of the current state-of-the-art of Systems Engineering (SysEng) practices. This document describes the SysEng concepts, and identifies the current enabling standards, processes, methods, frameworks and approaches used in engineering a system. In addition, a discussion is presented on the challenges in the acquisition of defence capabilities. Recommendations are identified for the Collaborative Capability Definition Engineering and Management Technology Demonstrator (CapDEM TD) initiative on the potential SysEng practices to be developed and implemented that can address these challenges within a Systems of Systems (SoS) environment.

1.2 Background

The DND/CF is currently engaged in an effort to fundamentally change its approach to planning. In the threat-based force-planning world of the past, requirements were often developed, validated, and approved as stand alone solutions, not as participating elements in an overarching SoS. This stovepipe approach to acquisition was not informed by, nor coordinated with, other elements of DND. Worse, the idea-to-capability cycle can be as long as 15 years.

The capabilities-based methodology that DND wishes to develop is intended to facilitate force planning in an uncertain environment by identifying the broad set of capabilities required for the next strategic planning cycle (out to approximately 2020). This shift away from threat-based to capability-based force structure planning will ultimately require a capability-based engineering approach based upon a sound SysEng methodology and SoS framework. This will allow for the 'step-wise' requirement-capability (spiral development) cycle, as well as the reduction by 30% of the long and costly acquisition cycle.

The current implementation of Capability-Based Planning (CBP) leads to the acquisition of platforms and systems without any formal method.

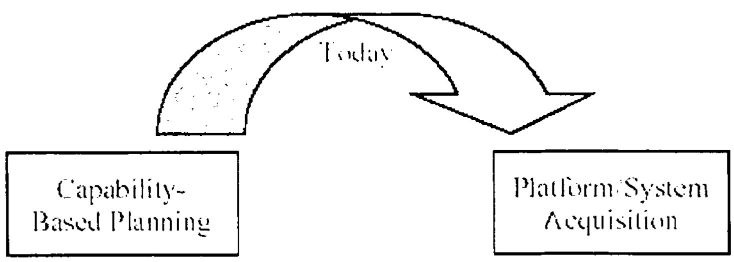

***Figure 1**: Capability-based planning - today [44]*

A systematic link between the conceptualization of a capability and the detailed definition of the component systems does not exist. Nor is there an analytical process or environment where trade-off analysis can be conducted across systems to evaluate their impact on each other or on the overall capability. In order to systematize this capability development and acquisition process, the rigour of a SysEng process is required [44].

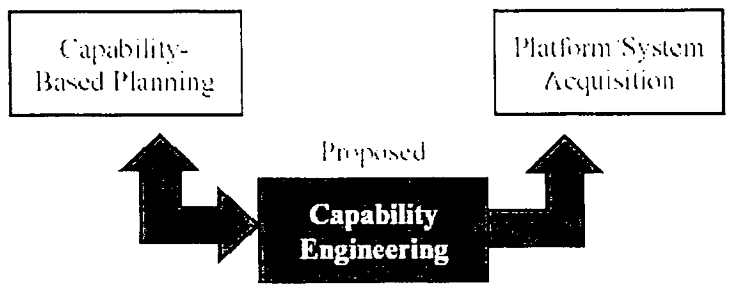

Figure 2: Capability-Based Planning - the proposed solution [44]

1.3 Systems Engineering Defined

The following are definitions of the terms and concepts related to SysEng that are discussed in this document.

1.3.1 Engineering

"Engineering is the application of a systematic, disciplined, quantifiable approach to structures, machines, products, systems, or processes" [1].

1.3.2 Systems

A system is defined as "a set or arrangement of elements (people, products (hardware and software) and processes (facilities, equipment, material, and procedures) that are related and whose behaviour satisfies customer/ operational needs, and provides for the life cycle sustainment of the products" [2].

The concept of a system is further described in ISO/IEC 15288-2002 as "a combination of interacting elements organized to achieve one or more stated purposes. These system elements are organized as a hierarchy of elements and systems. One person's system of interest can be viewed as a system element in another person's system of interest. A system may be considered a product or as the services it provides." [3].

The system paradigm is the foundation of SysEng. A system of interest can be viewed as an element of a larger system, and the challenge is to understand the boundary of the system of interest, which is the focus of the development

effort, and the relationships and interfaces among this system of interest to other systems. A system of interest is typically composed of several related elements and their interfaces. Additionally, elements include the people required to develop, produce, test, distribute, operate, support, or dispose of the element's products or to train those people to accomplish their role within the system of interest. Figure 3 provides a hierarchy of names for the elements making up a system of interest. The human elements are integral to the systems hierarchy, and may be present at any level. The human elements are not identified in the system hierarchy since the intent of the hierarchy is to identify the system element for which the system is being defined, and the human/system integration issues must be addressed in terms of the human's role in operating, producing and supporting the system.

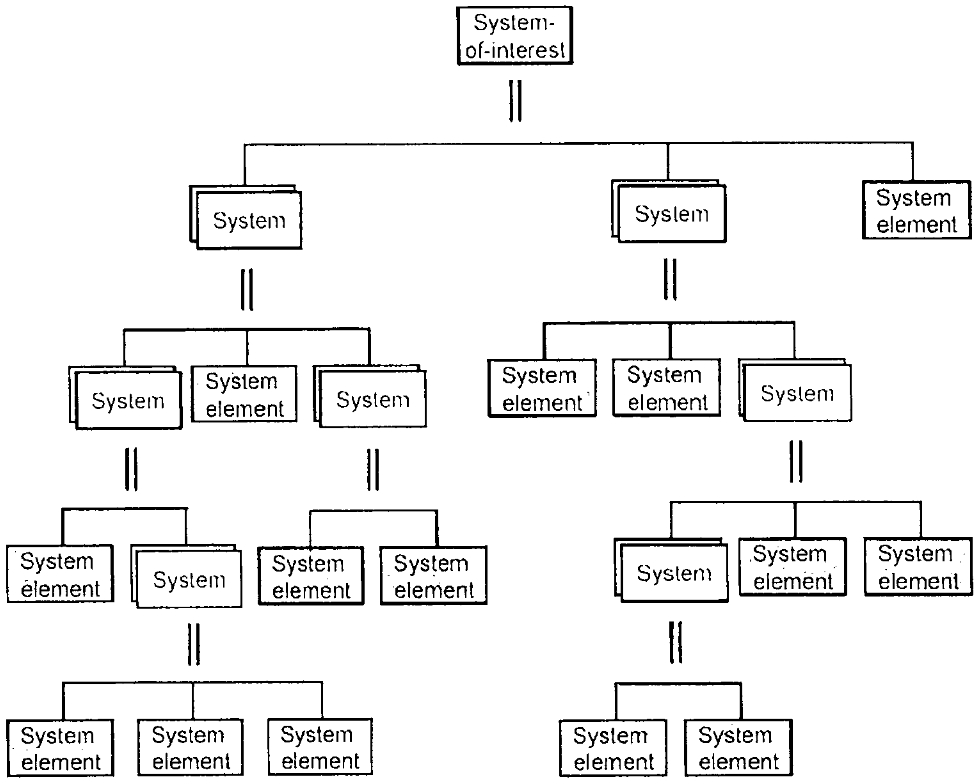

Figure 3: Hierarchy of elements within a system Systems Engineering [3]

1.3.3 Systems Engineering

Systems Engineering is "an interdisciplinary collaborative approach to derive, evolve and verify the life cycle balanced system solution to meet customer expectations and public acceptability"[4]. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem:

- Operations;
- Performance;
- Test;
- Manufacturing;
- Cost & Schedule;
- Training & Support; and
- Disposal.

"Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs" [6]. Figure 4 provides a view of the SysEng process and its role within the enterprise and external environments. The SysEng process is applied recursively during each phase of the full product life cycle. Initially, it is applied to identify the best concept, or approach, to satisfy the market opportunity. The Enterprise Environment defines the policies and procedures, determines the standards, the general specifications and guidelines, and then identifies the resources and domain technology within the constraints of the External Environment (political, economic, technological, etc). The Enterprise Environment decides on the projects to be executed. The Project Environment develops the plans, identifies the team, tools, controls and metrics to be used in the development of Systems.

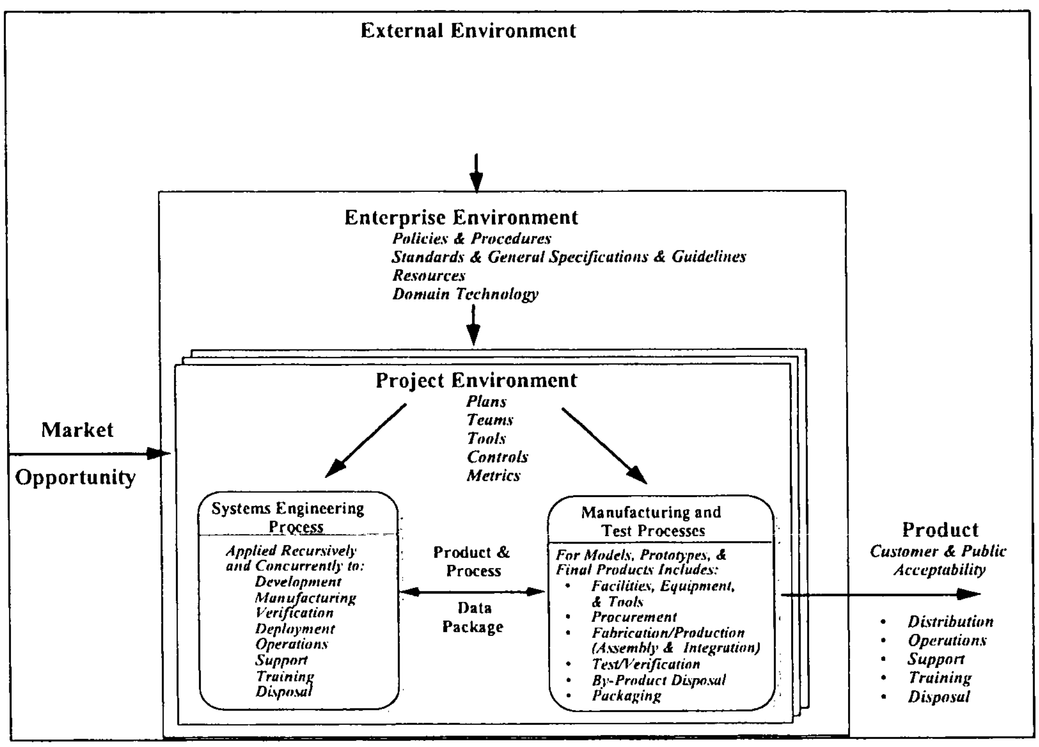

Figure 4: The systems engineering environment within an enterprise [3]

1.3.4 Process

"The process is the means by which people, procedures, methods, equipment and tools are integrated to produce a desired result" [9]

1.3.5 System Life Cycle

"System Life Cycle is the evolution over time of a system of interest from conception through to retirement" [3]. SysEng processes are applied to varying degrees throughout the Life Cycle of a system of interest. The names of the stages of a system life cycle differ between various standards. However the intent is to cover all phases of a system from 'cradle to grave'. A typical breakdown of the Life Cycle stages is as follows [3]:

- Conception;
- Development;
- Production;
- Utilization;
- Support; and
- Retirement.

1.4 Overview

SysEng processes are involved throughout the life cycle of a system of interest. As shown in Figure 5, once an enterprise has identified a need it will initiate various projects to further define the requirements, validate those requirements, conceive the appropriate system of interest that would best satisfy those requirements and then proceed with the development, production, utilization, support and eventually its retirement. The SysEng processes can be performed many times within each cycle to achieve the required objectives of that particular life cycle stage. Also depending on the life cycle stage, a selected set of SysEng processes are performed to satisfy the objectives of that life cycle stage. Therefore, each enterprise and, if determined necessary, each project, defines the set of SysEng processes to be performed during each life cycle stage that addresses the enterprises or the project's unique environment and satisfies the exit criteria for that stage. These SysEng processes are defined in various standards.

Various criteria are used to determine which processes to perform. Each organization is driven by the nature of its business and the environment within which it operates. These provide constraints and opportunities, which help form its policies and procedures that guide the selection of projects. Processes would be selected and tailored based on the enterprises policies, its culture, the project size and complexity and the technology involved.

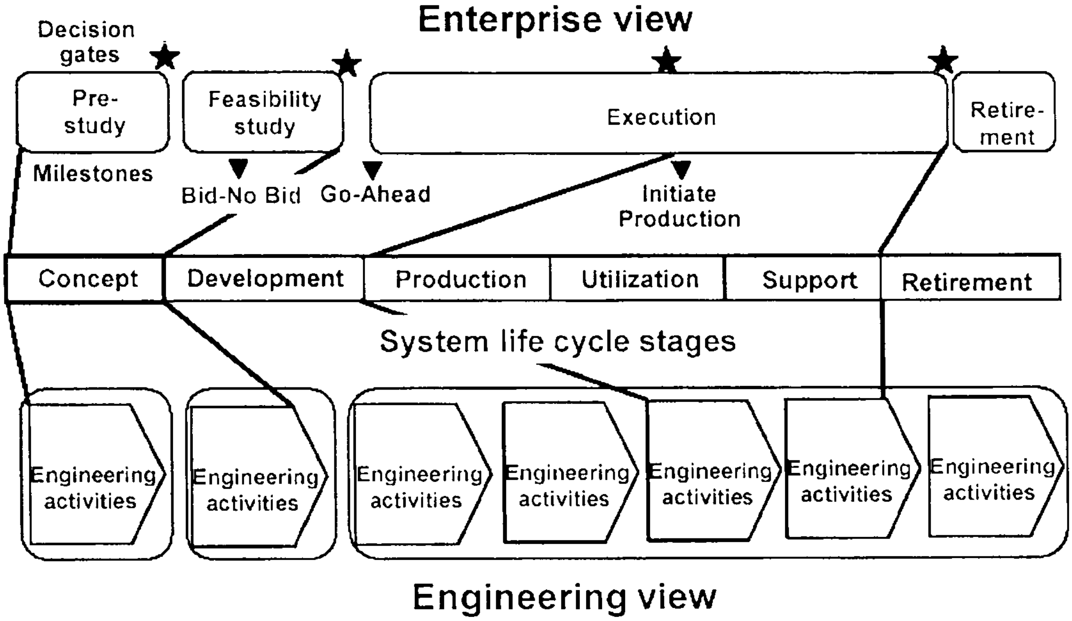

***Figure 5**: Enterprise and engineering views related to system life cycle stages [10].*

Various standards, models and frameworks are available that address SysEng. They provide varying levels of detail and address various domains. The standards can be grouped into three categories: SysEng Process Standards; Capability Maturity Models; and Domain-Specific Standards and Frameworks.

The current standards that address SysEng are the ISO/IEC 15288 Systems Engineering – System Life Cycle Processes [3], EIA 632 Processes for Engineering a System [6], and IEEE 1220 Standard for Application and Management of the Systems Engineering Process [4]. These standards differ in terms of scope and level of detail as shown in Figure 6. The ISO/IEC 15288 identifies the system life cycle processes for all system life cycle stages at a high level. The EIA 632 provides the next level of detail for the concept, development and production life cycle stages. The IEEE 1220 provides the greatest level of detail mainly covering the development stage [3].

Capability Maturity Models (CMM) [15] provide the framework to measure the effectiveness of, and to improve the SysEng processes of a particular organization or unit.

For a system that contains product/system elements for which lower-tier standards or frameworks exist, or where standards or guides exist for safety, security, or other system aspects, these are to be used in conjunction with the overarching SysEng standards. The Domain-Specific Standards and Frameworks include those standards and frameworks that provide the next level of detail. These standards are applied according to the nature of the system element to be developed. For example, to produce a system element that requires software creation the ISO/IEC 12207 Information

Technology – Software Life Cycle Processes [5]] and the EIA 649 Standard for Configuration Management [7] could be used.

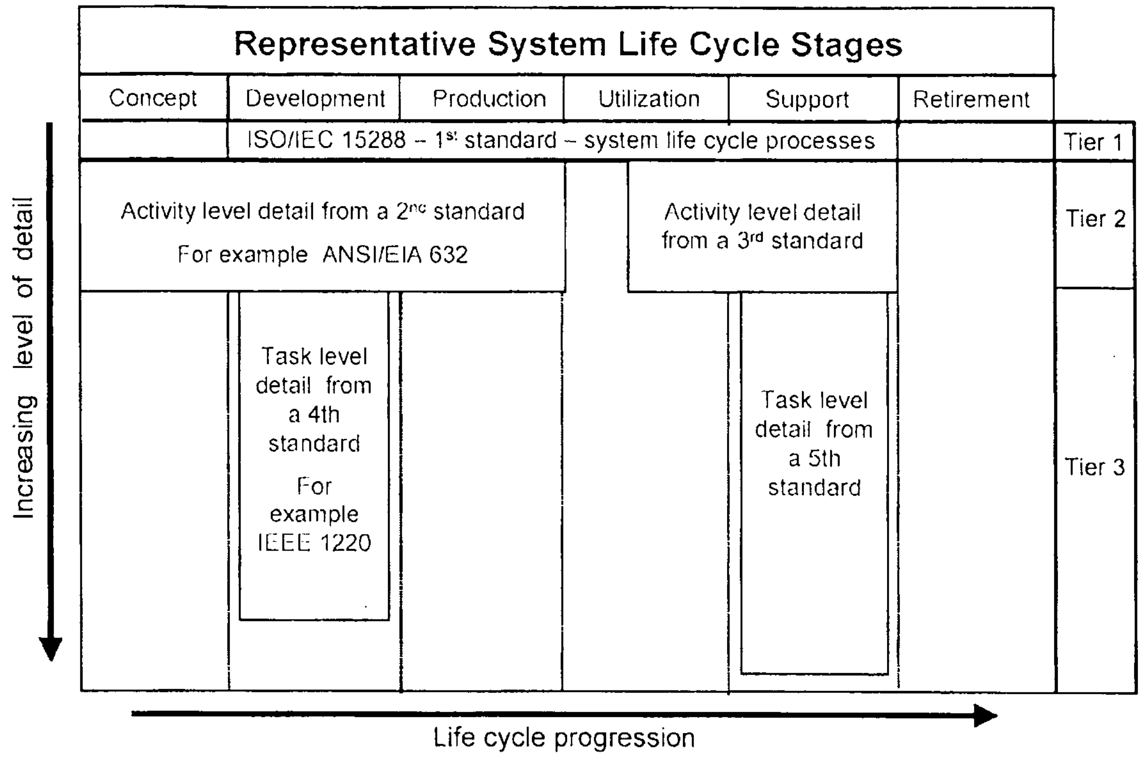

Figure 6*: Relationship of Systems Engineering standards to life cycle stages [3].*

To further understand the relationship between the three standards a mapping of the ISO 15288 system life cycle process groups to the EIA 632 and IEEE1220 process groups is shown in Table 1. From left to right in the table the breadth of coverage becomes narrower and the depth of coverage greater for the ISO 15288 system life cycle processes.

Table 1: *Mapping of ISO 15288 system life cycle process groups to the EIA 632 and IEEE 1220 engineering processes.*

ISO 15288	EIA 632	IEEE 1220
Enterprise Processes	Nil	Nil
Project Processes	Technical Management	Control
Technical Processes	System Design Product Realization Technical Evaluation	Requirements Analysis Requirements Verification Functional Analysis Functional Verification Synthesis Design Verification Systems Analysis
Agreement Processes	Acquisition Supply	Nil

2. Current Practices and Trends

Several standards embody the current practices in SysEng. The standards are grouped into three categories: SysEng Process Standards; Capability Maturity Models; and Domain-Specific Standards and Frameworks. As shown in Figure 7, these standards have evolved over several years and fortunately have merged to provide a consistent view within each category. The current SysEng Standards are the ISO/IEC 15288-2002 Systems Engineering – System Life Cycle Processes [3], EIA 632 Processes for Engineering a System [6], and IEEE 1220-1998 Standard for Application and Management of the Systems Engineering Process [2]. These standards are not equal in terms of scope and application. An overview of each standard is described below.

As mentioned above, the ISO/IEC 15288-2002 addresses the high level SysEng processes through the complete life cycle stages of a product or service being developed from conception to retirement. The EIA 632 provides the next level of detail mainly for the concept, development and production life cycle stages. The IEEE 1220-1998 provides the next level of detail of the SysEng process and specifically covers the development life cycle stage and does not cover the breadth of the life cycle as addressed in the other two standards.

Capability Maturity Models (CMM) [14] provide the framework to measure the effectiveness of the SysEng process of a particular organization or unit. The Capability Maturity Model® Integration (CMMI) [15] integrates several maturity models providing a single, aligned, and comprehensive model for assessing the whole SysEng process.

The Domain-Specific Standards and Frameworks provide the next level of detail and are applied according to the nature of the system element to be developed.

These standards do not specify the details of "how to" implement process requirements for engineering a system. Nor do they specify the methods or tools a developer would use to implement the process requirements. A systems engineer would need to select or define the methods and tools that are applicable to the development, and that are consistent with the enterprise policies and procedures. Several handbooks are identified here that do describe the processes to be applied for a particular industry.

In addition to the standards there are certain approaches that identify best practices in the application of these processes. The approaches discussed here are: sequential; incremental; evolutionary; agile; architecture-driven; modelling and simulation; concurrent engineering; spiral development; open systems; and product improvement.

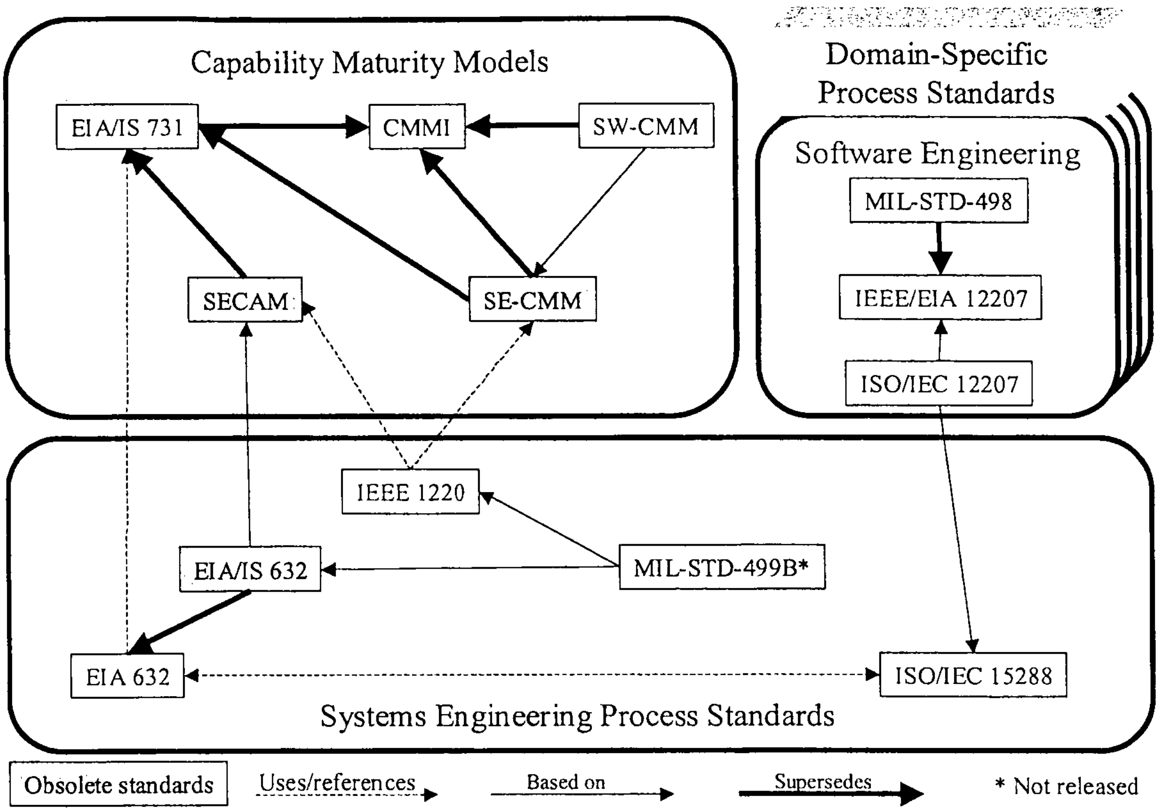

***Figure 7**: The frameworks quagmire [3]. [45]*

2.1 Systems Engineering Standards

2.1.1 ISO/IEC 15288 Systems Engineering – System Life Cycle Processes

This International Standard establishes a common framework for describing the life cycle of systems created by humans. This Standard concerns those systems that are man-made and may be configured with one or more of the following: hardware, software, humans, processes (e.g. review process), procedures (e.g. operator instructions), facilities and naturally occurring entities (e.g. water, organisms, minerals). This Standard defines a set of processes and associated terminology that can be applied to the full life cycle of systems, including conception, development, production, utilization, support and retirement of systems, and to the acquisition and supply of systems, whether performed internally or externally to an organization. The life cycle processes of this Standard can be applied concurrently, iteratively and recursively to a system and its elements. These processes can be applied at any level in the hierarchy of a system's structure. Selected sets of these

[3] The bottom part of this Figure is a legend explaining the diagramming elements used

processes can be applied throughout the life cycle for managing and performing the stages of a system's life cycle. This is accomplished through the involvement of all interested parties with the ultimate goal of achieving customer satisfaction.

This Standard has been adopted by the US DoD [43] and the UK MoD [26].

Each life cycle process may be invoked, as required, at any time throughout the life cycle and there is no definitive order in their use. Any life cycle process may be executed concurrently with any other life cycle process. Any life cycle process may apply at any level in the hierarchical representation of a system's structure. Therefore, in the following description of the system life cycle processes, the order that the processes are presented and the process groups used do not imply any precedence in, or sequence of application of, processes during the life cycle of a system.

The system life cycle processes are described in four process groups, each of which consists of subprocesses as shown in Figure 8 and described thereafter:

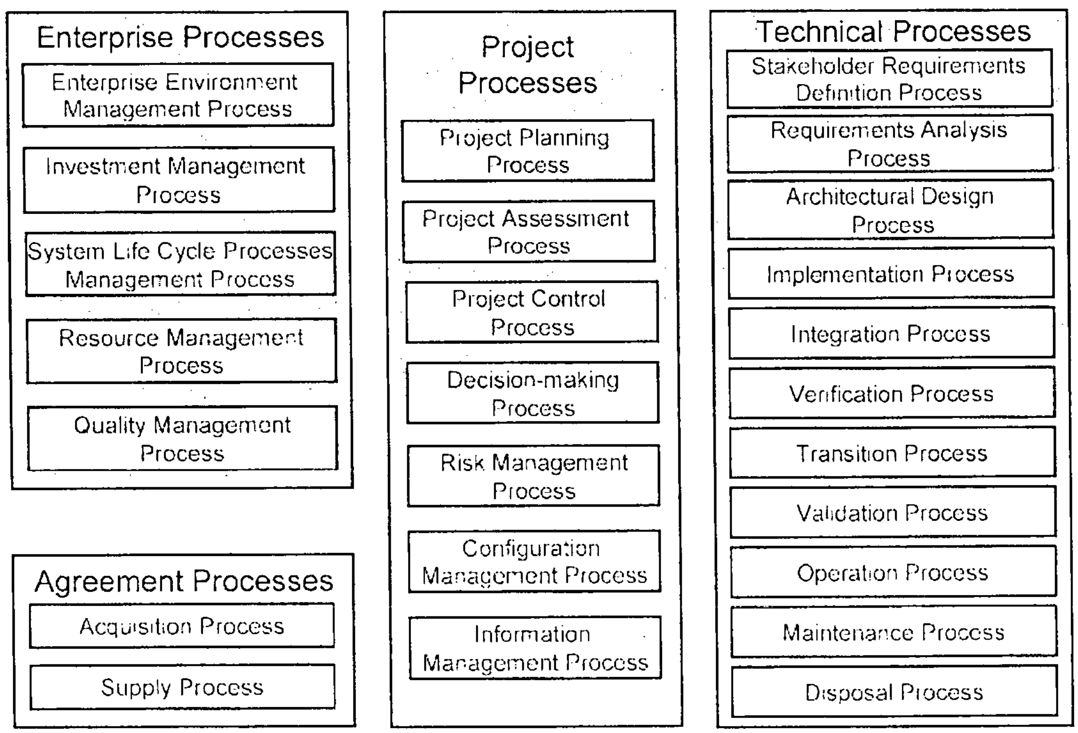

***Figure 8**: ISO/IEC 15288 Systems engineering system life cycle processes [3].*

Agreement processes

These processes specify the requirements for the establishment of agreements with organizational entities external and internal to the organization. The agreement processes are:

- Acquisition Process; and
- Supply Process.

Enterprise processes

These processes manage the organization's capability to acquire and supply products or services through the initiation, support and control of projects. They provide resources and infrastructure necessary to support projects and ensure the satisfaction of organizational objectives and established agreements. The enterprise processes are:

- Enterprise Environment Management Process;
- Investment Management Process;
- System Life Cycle Processes Management Process;
- Resource Management Process; and
- Quality Management Process.

Project processes

These processes are used to establish and evolve project plans, to assess actual achievement and progress against the plans and to control execution of the project through to fulfilment. The project processes are:

- Project Planning Process;
- Project Assessment Process;
- Project Control Process;
- Decision-making Process;
- Risk Management Process;
- Configuration Management Process; and
- Information Management Process.

Technical processes

These processes are used to define the requirements for a system, to transform the requirements into an effective product, to permit consistent reproduction of the product where necessary, to use the product to provide the required services, to sustain the provision of those services and to dispose of the product when it is retired from service. The technical processes are:

- Stakeholder Requirements Definition Process;
- Requirements Analysis Process;
- Architectural Design Process;
- Implementation Process;
- Integration Process;
- Verification Process;
- Transition Process;
- Validation Process;
- Operation Process;

- Maintenance Process; and
- Disposal Process.

2.1.2 Electronics Industries Alliance (EIA) 632: Processes for Engineering a System

This Standard provides an integrated set of fundamental processes to aid a developer in the engineering or reengineering of a system. This Standard is intended to enable an organization to improve its competitiveness in global markets by engineering and producing quality products and delivering them on time at an affordable price or cost. This Standard is a high-level standard, intended to apply to the engineering of any kind of system. The intended uses of this standard include:

- Benchmarking by an enterprise against the requirements of EIA 632 for engineering a system, or portion thereof;
- Preparing enterprise standards, policies and procedures for engineering a system;
- Developing lower-tier industry or domain specific process standards;
- Developing process capability and assessment models;
- Establishing terminology and concepts for better communications;
- Developing training and educational materials; and
- Preparing plans for actual development of products.

As shown in Figure 9 this Standard consists of a series of processes, in groups, with loops among them. The groups are described hereafter.

Acquisition and Supply Processes

These processes are used by the developer to arrive at an agreement with another party to accomplish specific work and to deliver required product, or to have work done and obtain desired product. The processes and the requirements covered by each process are as follows:

- Supply Process, that covers:
  - Product Supply.
- Acquisition Process, that covers:
  - Product Acquisition; and
  - Supplier Performance.

Technical Management Processes

These processes are used to plan, assess and control the technical work efforts required to satisfy the established agreement. The processes and the requirements covered by each process are as follows:

- Planning Process which covers:

- Process Implementation Strategy;
    - Technical Effort Definition;
    - Schedule and Organization;
    - Technical Plans; and
    - Work Directives.
- Assessment Process which covers:
    - Progress Against Plans and Schedules;
    - Progress Against Requirements; and
    - Technical Reviews.
- Control Process which covers:
    - Outcome Management; and
    - Information Dissemination.

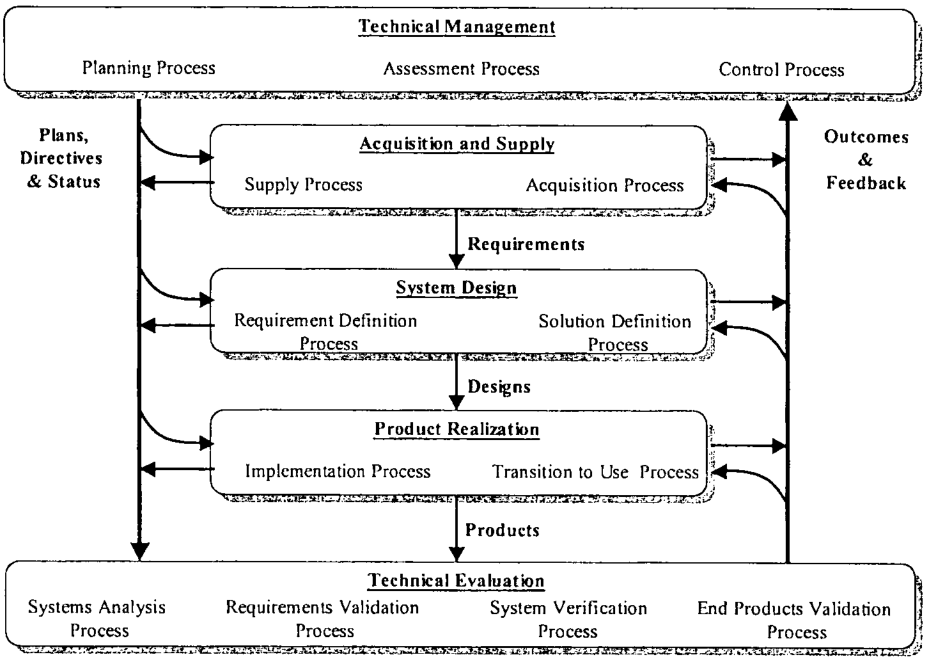

***Figure 9**: EIA 632 Relationship of processes for engineering a system [6]*

System Design Processes

These processes are used to convert agreed-upon requirements of the acquirer into a set of realizable products that satisfy acquirer and other stakeholder requirements. The processes and the requirements covered by each process are as follows:

- Requirements Definition Process which covers:
    - Acquirer Requirements;
    - Other Stakeholder Requirements; and
    - System Technical Requirements.
- Solution Definition Process which covers:
    - Logical Solution Representations;

- Physical Solution Representations; and
- Specified Requirements.

Product Realization Processes

These processes are used to (1) convert the specified requirements and other design solution characterizations into either a verified end product or a set of end products in accordance with the agreement and other stakeholder requirements; (2) deliver these to designated operating, customer, or storage sites; (3) install these at designated operating sites or into designated platforms; and (4) provide in-service support, as called for in an agreement. The processes are as follows:

- Implementation Process; and
- Transition to Use Process.

Technical Evaluation Processes

These processes are intended to be invoked by one of the other processes for engineering a system. The processes and the requirements covered by each process are as follows:

- System Analysis Process:
  - Effectiveness Analysis;
  - Trade-off Analysis; and
  - Risk Analysis;
- Requirements Validation Process:
  - Requirements Statements Validation;
  - Acquirer Requirements Validation;
  - Other Stakeholder Requirements Validation;
  - System Technical Requirements Validation; and
  - Logical Solution Representations Validation.
- System Verification Process:
  - Design Solution Verification;
  - End Product Verification; and
  - Enabling Product Readiness.
- End Products Validation Process.

2.1.3 IEEE 1220-1998 – Applications and Management of the Systems Engineering Process

This standard defines the interdisciplinary tasks, which are required throughout a system's life cycle to transform customer needs, requirements, and constraints into a system solution. This standard is intended for guiding the development of systems (which includes humans, computers, and software) for commercial, government, military, and space applications. This standard applies to an enterprise within an enterprise, which is responsible for developing a product design and establishing the life-cycle infrastructure needed to provide for life-cycle sustainment.

This standard specifies the requirements for the systems engineering process and its application throughout the product life cycle. The standard does not attempt to define the implementation of each system life cycle process, but addresses the issues associated with defining and establishing supportive life cycle processes early and continuously throughout product development. In addition, the standard does not address the many cultural or quality variables, which must be considered for successful product development. The standard focuses on the engineering activities necessary to guide product development while ensuring the product is properly designed to make it affordable to produce, own, operate, maintain, and eventually to dispose of, without undue risk to health, or the environment.

The requirements of this standard are applicable to new products as well as incremental enhancements to existing products. It applies to one-of-a-kind products, such as a satellite, as well as to products, which are mass-produced for the consumer marketplace. The requirements of this standard should be selectively applied for each specific system development project. The focus of this standard is product-oriented systems such as automobiles, airplanes, or information systems.

The Systems Engineering Process

The systems engineering process described in this standard is presented in Figure 10. The systems engineering process is a generic problem-solving process which provides the mechanisms for identifying and evolving the product and process definitions of a system. The SysEng process applies throughout the system life cycle to all activities associated with product development, test, manufacturing, training, operation, support, distribution, disposal, and human system engineering. Figure 10 depicts the elements of the SysEng process and shows how they iterate to produce a consistent set of requirements, functional arrangements, and design solutions. The details of these processes are described in the standard. A listing of the general flow of tasks for each of the subprocesses of the SysEng process is described here.

The SysEng process is applied throughout the system life cycle: development; manufacturing; test; distribution; operations; support; training; and disposal.

Each life cycle process is itself like a system in that products must be developed to fulfill the purpose of the life-cycle process.

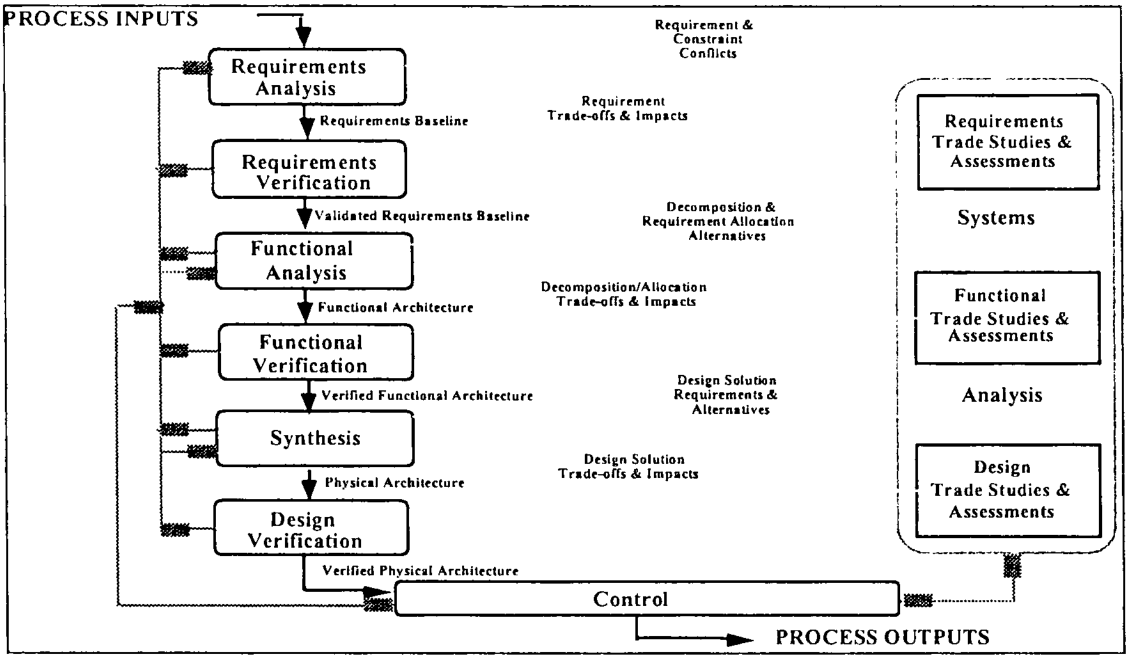

***Figure 10**: IEEE 1220-1998 System Engineering Process [2].*

The list of the tasks for each of the subprocesses of the Systems Engineering process is shown in Table 2 below.

***Table 2:** Systems Engineering Subprocesses*

REQUIREMENTS ANALYSIS	
• Define Customer Expectations	• Define Life-Cycle Process Concepts
• Define Project and Enterprise Constraints	• Define Functional Requirements
• Define External Constraints	• Define Performance Requirements
• Define Operational Scenarios	• Define Modes of Operation
• Define Measures of Effectiveness (MOE);	• Define Technical Performance Measures
• Define System Boundaries	• Define Physical Characteristics
• Define Interfaces	• Define Human Factors
• Define Utilization Environments	• Establish Requirements Baseline
REQUIREMENTS VERIFICATION	
• Compare to Customer Expectations	• Establish Validated Requirements Baseline
• Compare to Enterprise and Project Constraints	• Identify Variances and Conflicts
• Compare to External Constraints	
FUNCTIONAL ANALYSIS	
• Analyze Functional Behaviors	• Define Functional Timelines
• Define Functional Interfaces	• Define Data and Control Flows
• Allocate Performance Requirements	• Define Functional Failure Modes and Effects
• Define Subfunctions	• Define Safety Monitoring Functions
• Define Subfunction States and Modes	• Establish Functional Architecture

FUNCTIONAL VERIFICATION	
• Define Verification Procedures	• Verify Functional & Performance Measures
• Define Verification Evaluation	• Verify Satisfaction of Constraints
• Define Variances and Conflicts	• Identify Variances & Conflicts
• Verify Architecture Completeness	• Verified Functional Architecture
SYNTHESIS	
• Group and Allocate Functions	• Identify Make or Buy Alternatives
• Identify Design Solution Alternatives	• Develop Models and Prototypes
• Assess Safety and Environmental Hazards	• Assess Failure Modes, Effects, and Criticality
• Assess Life-Cycle Quality Factors	• Assess Testability Needs
• Assess Technology Requirements	• Assess Design Capacity to Evolve
• Define Physical and Performance Characteristics	• Finalize Design
• Define Physical Interfaces	• Initiate Evolutionary Development
• Identify Standardization Opportunities	• Produce Integrated Data Package
• Identify Off-The-Shelf Availability	• Establish Design Architecture
DESIGN VERIFICATION	
• Select Verification Approach	• Verify Satisfaction of Constraints
• Establish Verification Evaluation	• Verified Design Architecture
• Define Verification Procedure	• Verified Design Architecture of Life-Cycle Process
• Define Inspection, Analysis, Demonstration or Test Requirements	• Verified System Architecture
• Conduct Verification Evaluation	• Establish Specifications and Configuration Baselines
• Verify Architecture Completeness	• Develop Product Breakdown Structure
• Verify Functional And Performance Measures	
SYSTEMS ANALYSIS	
• Assess Requirement Conflicts	• Analyze Life Cycle Costs
• Assess Functional Alternatives	• Analyze and Costs Effectiveness
• Assess Solution Alternatives	• Analyze Environmental Impacts
• Identify Risk Factors	• Quantify Risk Factors
• Define Trade Study Scope	• Select Risk Handling Option
• Select Methodology and Success Criteria	• Select Alternative Recommendation
• Identify Alternatives	• Trade-offs and Impacts
• Establish Trade Study Environment	• Design Effectiveness Assessment
• Conduct Trade Study	
CONTROL	
• Technical Management	• Track Performance Against Project Plans;
• Data Management	• Track Performance Against Technical Plans
• Configuration Management	• Track Product and Process Metrics
• Interface Management	• Update Specifications and Configuration Baselines
• Risk Management	• Update Requirements Views and Architectures
• Performance Based Programs Management	• Update Engineering Plans
• Track System Analysis and Verification/Test Data	• Update Technical Plans
• Track Requirement and Design Changes	• Integrated Database

2.2 Capability Maturity Models

The purpose of a Capability Maturity Model (CMM) [15] is to provide guidance for improving an organization's processes and their ability to manage the development, acquisition, and maintenance of products or services. As shown in Figure 7, several CMMs have been developed over time to address different domains such as Software Engineering, Systems Engineering, Integrated Product and Process Development, and Supplier Sourcing. Several of these models (e.g. EIA/IS731, SW-CMM) have been superseded by CMM Integration (CMMI) [14] which aims to provide an integrated approach across the enterprise for improving processes, while reducing the redundancy, complexity and cost resulting from the use of separate and multiple CMMs.

CMMI places proven approaches into a structure that helps an organization appraise its organizational maturity or process area capability, establish priorities for improvement, and implement these improvements. CMMI is compatible and conformant with ISO/IEC 15504 International Standard for Software Process Assessment [46].

The CMMI Product Suite contains and is produced from a framework that provides the ability to generate multiple models and associated training and appraisal materials. These models may reflect content from bodies of knowledge (e.g., systems engineering, software engineering, Integrated Product and Process Development) in combinations that are most useful (e.g., CMMI-SE/SW, CMMI-SE/SW/IPPD/SS).

"The CMMI models improve upon the best practices of previous models in many important ways. CMMI best practices enable organizations to do the following: [16]

- More explicitly link management and engineering activities to business objectives;
- Expand the scope of and visibility into the product life cycle and engineering activities to ensure that the product or service meets customer expectations;
- Incorporate lessons learned from additional areas of best practice (e.g., measurement, risk management, and supplier management);
- Implement more robust high-maturity practices;
- Address additional organizational functions critical to its products and services; and
- More fully comply with relevant ISO standards."

Organizations, both commercial and defence related, use CMMI to structure their process improvement efforts. These models significantly influence investments in process improvement activities. The SysEng models are being used by mature organizations engaged in complex engineering activities. CMMI is routinely used for benchmarking of organization practice and for establishing process improvement plans.

The initial model set of the CMMI product suite contains the following:

- Software Engineering (CMMI-SW);
- Systems Engineering (CMMI-SE);

- Systems Engineering + Software Engineering (CMMI-SE/SW); and
- Systems Engineering + Software Engineering with Integrated Product & Process Development (CMMI-SE/SW/IPPD).

The CMMI product suite was developed specifically for those users who are system and product developers and want to improve their processes and products. Tools and models are provided that enable users to assess where they are, identify goals for future improvements, follow models of best practices to achieve those goals and use CMMI products to conduct training, perform assessments and do tailoring.

The CMMI Appraisal product is called Standard CMMI Appraisal Method for Process Improvement (SCAMPI). SCAMPI is designed to provide benchmark quality ratings relative to CMMI models. It is applicable to a wide range of appraisal usage modes, including both internal process improvement and external capability determinations. SCAMPI supports the conduct of ISO/IEC 15504 assessments as well.

2.3 Domain-Specific Standards and Frameworks

There are many standards and frameworks that cover engineering disciplines in greater detail and provide additional guidance in executing a SysEng Process. Table 3 below shows a sampling of some of these standards and frameworks:

Table 3: *Sample domain-specific standards and frameworks.*

DOMAIN	STANDARD
Software Engineering	• ISO/IEC 12207 Information Technology – Software life cycle processes; • IEEE 12207 Information Technology – Software life cycle processes;
Configuration Management	• EIA 649 Standard for Configuration Management • ISO/IEC 15846 Information Technology – Software Life Cycle Processes - Configuration Management
Quality	• ISO 9000 series
Project Management	• Project Management Institute Project Management Book of Knowledge (PMBok) [17] • ISO/IEC 16326 Software Engineering – Guide to the application of ISO/IEC 12207 to Project Management
Architecture	• US DoD Architecture Framework v1.0 • IEEE 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems • Canadian Treasury Board Secretariat Business Transformation Enablement Program (BTEP) [18]
Requirements	• IEEE 1233 Guide for Developing System Requirements Specifications
Software Maintenance	• IEEE 1219 Software Maintenance • ISO 14764 Information Technology – Software Maintenance
System Definition	• IEEE 1362 Guide for Information Technology – System Definition – Concept of Operations Document
Software Measurement	• ISO/IEC 15939, Software Engineering - Software Measurement

DOMAIN	STANDARD
	Process
Interoperability	• ISO AP-233 Systems Engineering – Data Representation and Exchange Standardisation (still under development) • XML Metadata Interchange (XMI) [19] • Levels of Information Systems Interoperability (LISI) [42]
Modelling	• Unified Modelling Language (UML) [20] • System Modelling Language (SysML [21]

2.4 Systems Engineering Handbooks and Guides

Several examples of SysEng handbooks and guides exist that address different industry sectors and provide valuable information on the application of the SysEng Processes within these domains.

2.4.1 ISO/IEC 19760 Systems Engineering - A Guide for the Application of ISO/IEC 15288 (System Life Cycle Processes)

The intent of this document is to provide guidance for the application of ISO/IEC 15288 to systems and projects of various size and types. It describes in greater detail the system and system life cycle concepts, and the application of the system life cycle processes. In addition, different approaches that can be used in applying the Standard are described.

2.4.2 NASA Systems Engineering Handbook

This handbook describes SysEng, as it should be applied to the development of major National Aeronautics and Space Administration's (NASA) systems. It provides general concepts and generic descriptions of processes based on an early version of the EIA 632 standard, tools and techniques [6]. It also provides information on good SysEng practices, and pitfalls that should be avoided. However, this handbook is dated and does not fully represent NASA's current practice. An update is planned [22].

2.4.3 UK MoD Defence Systems Engineering Handbook

This handbook [23] is a short guide that applies SysEng to the UK defence environment. It is a review of defence SysEng as taught by the Defence Engineering Group (DEG) at University College London for the UK MoD. It describes the differences between SysEng for civil applications as opposed to defence applications. It also describes the role that Defence Systems Engineering plays in the acquisition of defence systems as part of the UK

Smart Acquisition program described below. The SysEng Process is based on the ISO/IEC 15288 systems lifecycle standard [3].

2.4.4 UK MoD Acquisition Management System

The UK MoD Acquisition Management System is a comprehensive information site embodying the principles of Smart Acquisition (SA). It has been released on the Internet [26] and provides a wealth of information on the UK MoD Smart System Acquisition program. This program follows a 'through-life' approach, which is based on the ISO/IEC 15288 system life cycle standard [3]. This site is updated monthly.

2.4.5 INCOSE Systems Engineering Handbook

This handbook [8] provides a description of the key SysEng process activities that is applicable to most engineering projects. The purpose for each process activity, what needs to be done, and how it can be done is described in some detail. Enough information is provided in order to determine whether a process activity is appropriate in supporting the objectives of the program or project they support, and how to go about implementing the process activity. This handbook has been written to serve as a stand-alone reference for SysEng processes in conformance with EIA-632 and the related EIA/IS 731 Systems Engineering Capability Model [47]. In particular, this handbook compares commercial practices to the US DoD and presents a mapping of SysEng to the US DoD system life cycle phases.

2.4.6 Integrated Systems and Software Engineering Process (ISSEP)

This document describes the Integrated Systems and Software Engineering Process (ISSEP) created by the Software Productivity Consortium (the Consortium) [12]. ISSEP's purpose is to enable improvement of the overall systems development process allowing systems and software engineers to more efficiently perform their work. The focus of this document is on the development system life cycle stage. The ISSEP model defines a set of management and technical activities, and most importantly, defines the mechanisms to coordinate and control the development effort. The ISSEP model is intended to comply with all the major SysEng standards and provides a high-level process framework for implementing these standards, by defining activities and information flows for integrating the requirements documented in these standards.

2.4.7 US DoD Systems Engineering Fundamentals

As described in the preface of this book [41], it provides a basic, conceptual-level description of engineering management disciplines that relate to the

development and life cycle management of a system. For the non-engineer it provides an overview of how a system is developed. For the engineer and project manager it provides a basic framework for planning and assessing system development.

2.5 Systems Engineering Approaches

The trend in SysEng for highly complex systems is towards a holistic, through-life approach to which the above SysEng processes are applied following one, or a combination, of the approaches below.

The selection, development and use of various SysEng approaches depend on several factors such as the following [10]:

- "Acquisition policy of the organization;
- The nature and complexity of the system;
- The stability of system requirements;
- Technological opportunities;
- The need for different system capabilities at different times; and
- The availability of resources"

The SysEng approaches discussed here are sequential, incremental, evolutionary, agile, and architecture-driven. These approaches are not mutually exclusive and can be combined to address specific enterprise and project environments.

In combination with these approaches other techniques can be used such as modelling and simulation (M&S), concurrent engineering, spiral development, open systems approaches and product improvement strategies. These are also briefly described below.

2.5.1 Sequential Approach

A sequential approach (also know as the 'waterfall' approach) follows an orderly progression through the life cycle processes through each of the life cycle stages as shown in Figure 5.

This approach typically is used for systems that have the following characteristics [10]:

- "Long development cycles (five or more years);
- One of a kind (such as a particular model of automobile) produced in large quantities (such as in infrastructure information technology systems, automobiles, control systems and consumer products);
- Utilization and support stages are typically the longest stages; and
- Operational life of ten or more years with modifications using technology updates to sustain and lengthen useful life"

Several risks are inherent to this approach due to the long development cycle [10]:

- "Requirements and expectations may change;
- Personnel turnover including knowledge workers, decision makers and customers;
- Technology change, and obsolescence; and
- Suppliers go out of business".

There are also some opportunities with this approach [10]:

- "All capabilities delivered at the same time; and
- Old systems can be withdrawn from service simultaneously with the introduction of the new system".

2.5.2 Incremental Approach

The incremental approach delivers a number of versions of a system with increasing capability. The first version is allocated a limited set of functionality with each successive version adding more capability until the final version fully incorporates the overall capability.

This approach typically is used for systems that have the following characteristics [10]:

- "Systems that require updated capabilities frequently (such as medical systems, firewalls and automated business systems)"

Several risks are inherent to this approach [10]:

- "Customers could become dissatisfied due to limited capabilities of the early version;
- Customers may not be interested in paying for upgrades, if applicable;
- Training costs may be unacceptable with the new versions;
- May be required to support more than one version operationally;
- Configuration control;
- Unplanned technology change could significantly impact cost and schedule of new versions; and
- Requirements may change".

Some opportunities associated with this approach are as follows:

- "Acquirer requirements for early capabilities can be satisfied;
- Early prototypes can potentially be deployed operationally; and
- Early introduction of the system could win stakeholder buy-in"

2.5.3 Evolutionary Approach

The evolutionary approach is similar to the incremental approach in that it delivers a number of versions of a system with increasing capability. The only difference is that the full capabilities of the final version are unknown when system development is begun. The initial requirements are only

partially defined and refined with each successive version of the system that include lessons learned from the previous version.

This approach typically is used for systems that have the following characteristics [10]:

- "Complex systems where the requirements are not fully understood such as in military information technology systems"

Several risks are inherent to this approach [10]:

- "Customers could become dissatisfied due to limited capabilities of the early version;
- Training costs may be unacceptable with the new versions;
- May be required to support more than one version operationally;
- Configuration control;
- Unplanned technology change could significantly impact cost and schedule of new versions; and
- Requirements may change"

Some opportunities associated with this approach are as follows [10]:

- "Acquirer requirements for early capabilities can be satisfied;
- Customer feedback and lessons learned used to guide development of next version;
- Can take advantage of emerging technologies;
- Early prototypes can potentially be deployed operationally; and
- Early introduction of the system could win stakeholder buy-in"

2.5.4 Agile Systems Approach

"An agile systems approach has grown out of software engineering and is characterized by swift adaptation to changes, non-hierarchical baseline management, and a notable absence of low-value bureaucracy" [32]. "System engineers are encouraged to use 'agile methods' to develop systems that are 'agile performers', and have inherent 'agility,' meaning they can easily and rapidly adapt to continually changing requirements" [32].

This approach typically is used for systems that have the following characteristics [32]:

- "Few non-complex interfaces (as the number and complexity of interfaces increases so does the need for extensive schedule and documentation rigour);
- Homogenous set of stakeholders (a diverse set of stakeholders with competing interests can impede close collaboration and force a document-facilitated approach);
- Highly competent development team; and
- Effective communications within the development team"

Several risks are inherent to this approach [32]:

- "Customers could become dissatisfied as verbally agreed to features are not satisfactorily addressed;
- Long-term maintenance of system could be a problem without proper documentation;
- Subsequent refinements to the requirements are not necessarily captured; and
- Schedule and cost risk due to facility in changing requirements"

Some opportunities associated with this approach are as follows [32]:

- "Can quickly adapt to ever-evolving user requirements; and
- Can quickly adapt to changing and emerging technologies"

The following are Agile SysEng best practices as identified in [32]:

- "The project team understands, respects and works (behaves) within a defined SysEng process. (The process is systemic in the organization and implicit to the participants.)
- The project is executed as fast as possible with minimum down time or staff diversion to other priorities during the project. (Every opportunity is exercised to move the project forward, especially for the critical path activities, with minimal interruptions.)
- All key players are physically or virtually [by electronic links] collocated. Other contributors are available on-line 24x7. (High-bandwidth communication is rapid and effortless.)
- A strong bias for automatically generated electronic documentation. Engineers rely on their tools and their "Electronic Engineering Notebooks" to record decision rationale. Artefacts and documentation for operations and replication is done only if necessary—not to support an existing bureaucracy or policy. (Notebooks are team property and are available to all.)
- Baseline management and change control by formal, oral agreement based on "make a promise, keep a promise" discipline—players hold each other accountable. Control gates are settled with a handshake. (Formality relates to the reliability of the action not the amount of prescribed paperwork.)
- Opportunity exploration and risk reduction are accomplished by expert consultation and rapid model verification, coupled with close customer collaboration. Software development is done in a rapid development environment, while hardware is developed in a multi-disciplined model shop. (There is no resistance or inertia to securing expert help; it is sought rather than resisted.)
- A culture of constructive confrontation pervades the project organization. Issues are actively sought. Anyone can identify an issue and pass it on to the most likely solver. No issue is left unresolved. (The team takes ownership for success; it is never 'someone else's responsibility)."

2.5.5 Architecture-Driven Approach

Architecture-Driven Approach is the application of an architecture framework such as the DoD Architecture Framework to the Systems Engineering processes.

> *"The architecture is the first level of design that can be reasoned about. It provides the framework for analyzing both the engineering development and operational uses of the system" [33].*

As suggested by Dickerson et al. in [33], the four steps of a standard SysEng process (i.e., requirements analysis, functional analysis, synthesis, and design verification) can be associated with the following four product groups from the DoDAF version 1.0 (see Figure 11):

- Operational Concept;
- System Functional Mapping;
- System Interface Mapping; and
- Architecture Performance and Behaviour.

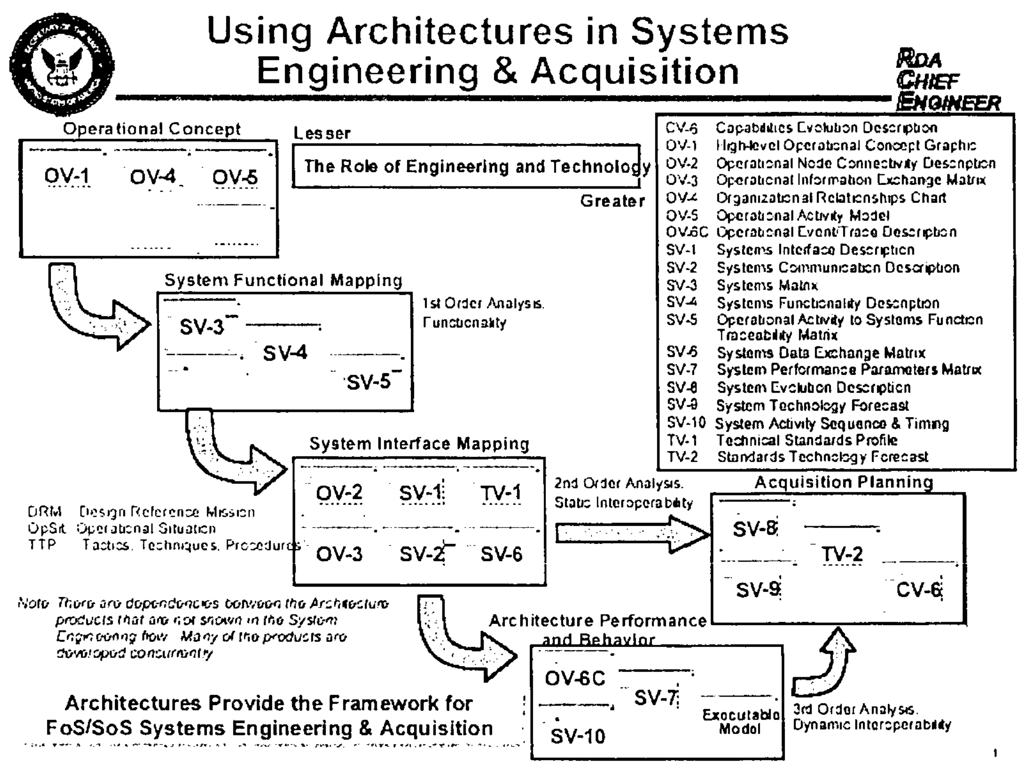

***Figure 11**: Using the DoD Architecture Framework in Systems of Systems Engineering and Acquisition [52, Slide 4]*

This approach typically is used for systems that have the following characteristics:

- Highly complex systems with a high number of, and complexity of interfaces, such as military information systems including Family-of-Systems and Systems of Systems;
- Diverse set of stakeholders;
- Geographically dispersed project teams; and
- Technology agnostic;

However, stakeholders may become dissatisfied due to longer time spent in the requirements to design phases before moving to development.

Opportunities associated with this approach are as follows:

- Able to address a diverse set of stakeholders as many views can be constructed to satisfy the needs and roles of each stakeholder;
- Able to address interoperability between systems as architecture provides a better understanding of the system interaction requirements; and
- Able to address re-usability of components between systems.

2.5.6 Modelling and Simulation

Modelling can be used to define the architectural views that represent the system characteristics such as operational business processes, information, and human-systems interaction. Model-Driven Architecture [34] is a specific technique used to accomplish this. These models can depict the dynamic behaviour of a conceptual system that can then be simulated in a synthetic simulation environment allowing for the refinement and robust and thorough validation of the requirements before moving on to development. UML-based Architecture descriptions can be directly reused in simulation environments. An in-depth discussion of the modelling and simulation roles is provided in Drouin and Wellwood [35].

2.5.7 Concurrent Engineering

Concurrent Engineering is a management/operational approach which aims to improve product design, production, operation, and maintenance by developing environments in which personnel from all disciplines (design, marketing, production engineering, process planning, and support) work together and share data throughout all phases of the product life cycle [30].

2.5.8 Spiral Development Method

The Spiral Development Model, as a risk-driven process model, can be used to choose the approach to be applied [13]:

> *"The spiral development model is a risk-driven process model generator. It is used to guide multi-stakeholder*

concurrent engineering of software intensive systems. It has two main distinguishing features. One is a cyclic approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk. The other is a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions. Figure 12 below illustrates the Spiral Development Process as originally proposed by Boehm".

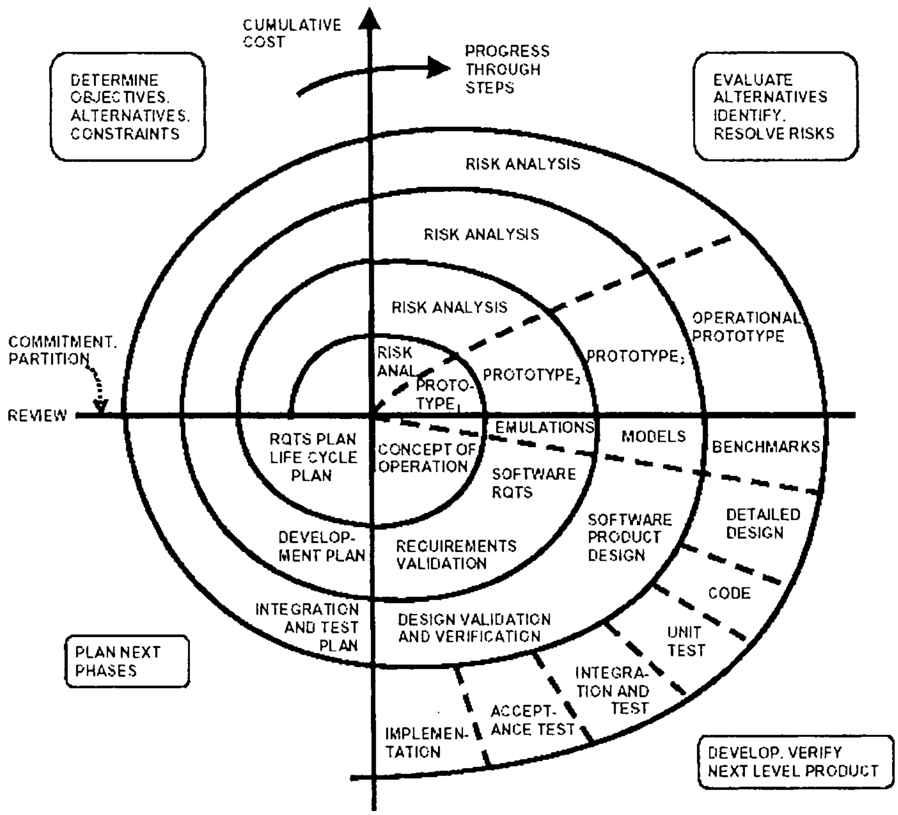

***Figure 12**: Barry Boehm's Original Spiral Development Model [48]*

2.5.9 Open Systems Approach

As described in [31], an Open Systems Approach is an integrated business and technical strategy that employs a modular design and, where appropriate, defines key interfaces using widely supported, consensus-based standards that are published and maintained by a recognized industrial standards organization.

The foundation of an Open Systems Approach is a modular design; isolating functionality makes the system easier to develop, maintain, and modify or upgrade. Modular designs are characterized by the following:

- Functionally partitioned into discrete scalable, reusable modules consisting of isolated, self-contained functional elements;
- Rigorous use of disciplined definition of modular interfaces, to include object oriented descriptions of module functionality; and
- Designed for ease of change to achieve technology transparency and, to the extent possible, makes use of commonly used industry standards for key interfaces.

The interfaces are key; interface standards specify the physical, functional, and operational relationships between the various elements (hardware and software), to permit interchangeability, interconnection, compatibility and/or communication. Interface standards must be well defined, mature, widely used, and readily available. In general, popular open standards yield the most benefit to the customer in terms of ease of future changes to the system and should be the standards of choice in most cases.

Adhering to standards is critical in enabling an open systems approach. Standards should be selected based on maturity, market acceptance, and allowance for future technology insertion.

2.5.10 Product Improvement Strategies

In the Systems Engineering Fundamentals document [41] several product improvement strategies are described. Systems need to be developed with an appreciation for future requirements, foreseen and unforeseen, which manifest themselves as needed upgrades to safety, performance, supportability, interface compatibility, or interoperability; changes to reduce cost of ownership or major rebuild. The strategies or design approaches that reflect the improvement needs are categorized as planned improvements, changes in design or production, and deployed systems upgrades.

Planned improvement strategies include evolutionary acquisition, pre-planned product development and open systems.

Changes in design or production address the orderly management of change through engineering change proposals and the grouping of a number of changes to be applied consistently as a single block.

Deployed systems upgrades address to ***major rebuild*** and ***post-production improvement*** scenarios. ***Major rebuilds*** are to be considered as engineering a new system and require the full developmental considerations of the SysEng process, base lining, and life cycle integration. ***Post-production improvement*** involves improving or maintaining the system as its components reach obsolescence. These improvements are characterized by an upgrade to component or a subsystem as opposed to a total system upgrade.

3. DND/CF Challenges

The challenges described hereafter are either of generic nature, applying to DND/CF as well as to most military organizations (e.g. US, UK, Australia), or specific to DND/CF as reported in our analysis of the current situation.

3.1 Generic Challenges

3.1.1 Military vs. Commercial Context

As described in Defence Engineering Group's Defence Systems Engineering Handbook [23], major defence projects tend to exhibit the following range of attributes:

- "It is common for defence projects to be large and expensive (such as aircraft and ships);
- Defence projects are often at the leading edge of technology to provide the capability to meet a high technology threat; hence they are inherently more advanced with higher risks;
- Defence projects generally have a long life from concept to disposal. This means that all the future benefits of a defence project are difficult to assess with confidence, particularly within the changing defence environment;
- The benefits of a defence project are difficult to express in financial terms;
- Defence projects are procured with public funds and, as such, are constrained by the necessity of public accountability and by political considerations;
- Defence projects have many interested stakeholder organisations (Armed Forces, contractors, allies, taxpayers etc.) all of which have an interest in the project. Stakeholders may have divergent views on what constitutes success and how it should be achieved (i.e. the armed forces seek capable equipment, contractors are looking for profits, allies expect complementary capabilities and taxpayers require responsible spending of public money);
- Defence equipment is often deployed rapidly and is expected to be operational and integrated with other local equipment/forces with limited setting up."

"Whilst it is acknowledged that civil projects share some of these attributes (a large bridge project would be expensive, medical breakthroughs can be high-tech, local government is constrained by budgets etc) defence projects are almost unique in that they display the full set of these characteristics." [23]

It is easy to verify that a very formal and rigid SysEng process characterize military systems. The drivers to those SysEng processes are [24]:

- "High Complexity (e.g. Systems of Systems);
- High Technology Risk (e.g. Development of UAVs);
- High Costs: (e.g. development, testing and evaluation);
- Extreme Design Constraints (e.g. 99% + reliability is mandatory);
- Thoroughness (i.e. Emphasis on perfectly workable solutions rather than costs);
- Auditability (e.g. the need to comply with Treasury Board policy); and
- Changing Political Environment (e.g. end of cold war, changing nature of threats, increased use of joint and combined forces, etc.)."

The major challenges that emerge from this rigid and complex SysEng environment are [25]:

- "The focus is shifting from platforms to capabilities solutions;
- System complexity is constantly increasing;
- Demand for network-centric capability drives higher levels of integration;
- Functional and physical Interfaces are expanding in number and complexity;
- Evolutionary Acquisition is institutionalizing change;
- System and technology life cycles are desynchronized (e.g. 30 to 40 years for systems, vs. less than five years for commercial technology);
- Requirement for system architectures that have the ability adapt to a rapidly changing external environment; and
- Requirement for Engineering Specialists to adapt to the changing nature of Systems (e.g. Systems of Systems)."

As stated in UK Defence Systems Engineering Handbook [23],

> *"Defence Systems Engineering is challenging because there are no universal rules. It requires a philosophical approach, treating each new problem in a rigorous manner, and using Systems Engineering principles to formulate suitable solutions. There are no pre-determined rules to finding solutions to these problems, but each problem must be resolved within the resources available and bearing in mind any constraints. Since the resources and constraints will vary from project to project, the problems must be examined anew and solutions cannot necessarily be transported between projects."*

It follows that the need for SysEng in Defence projects is even greater than in similar-sized civil projects. However, the specific SysEng discipline, which is required for Defence projects, is known as Defence SysEng (DSE) and defined by the Defence Engineering Group [23] as:

> *"The integration of those engineering, analytical and management activities necessary for the procurement of*

large and complex defence systems. It uses systems engineering philosophies and procedures to promote the achievement of performance, cost and timescale targets in the uncertain environment of rapidly advancing technology and evolving industrial and geo-political circumstances."

3.1.2 Capability vs. Systems Focus

The opportunity to use the SysEng process for Capability Management stems from deficiencies that were identified within DND/CF regarding the way capabilities defined by capability-based planning are actually acquired. CapDEM TD sponsors have defined these deficiencies as:

- The absence of systematic link between the conceptualization of a capability, as defined by capability-based planning, and the detailed definition of the components systems; and
- The absence of analytical process and environment where trade-off analysis can be conducted across systems to evaluate the impact on each other or on the overall capability.

In order to address these deficiencies, SysEng was envisioned as the key element that would bring the necessary rigour to DND/CF current acquisition process and environment.

The following definition of "Capability Engineering", shown in Figure 13, illustrates the goal pursued by CapDEM TD with regard to SysEng. The main issue, with regard to this opportunity, resides in the fact that SysEng has historically been applied to the development of individual Systems, whereas Capability Engineering is rather concerned by federations of systems commonly known as Systems of Systems (SoS).

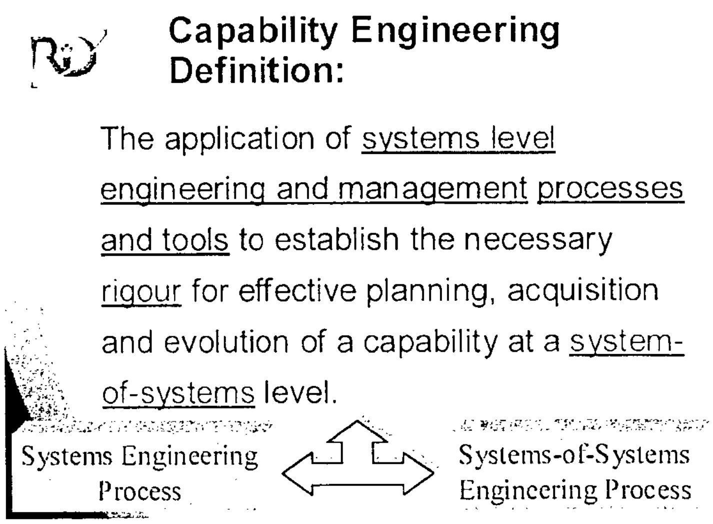

Figure 13: CapDEM TD Definition of Capability Engineering [49, Slide 17]

Figure 14 illustrates the relative position of SoS in the continuum of System complexity.

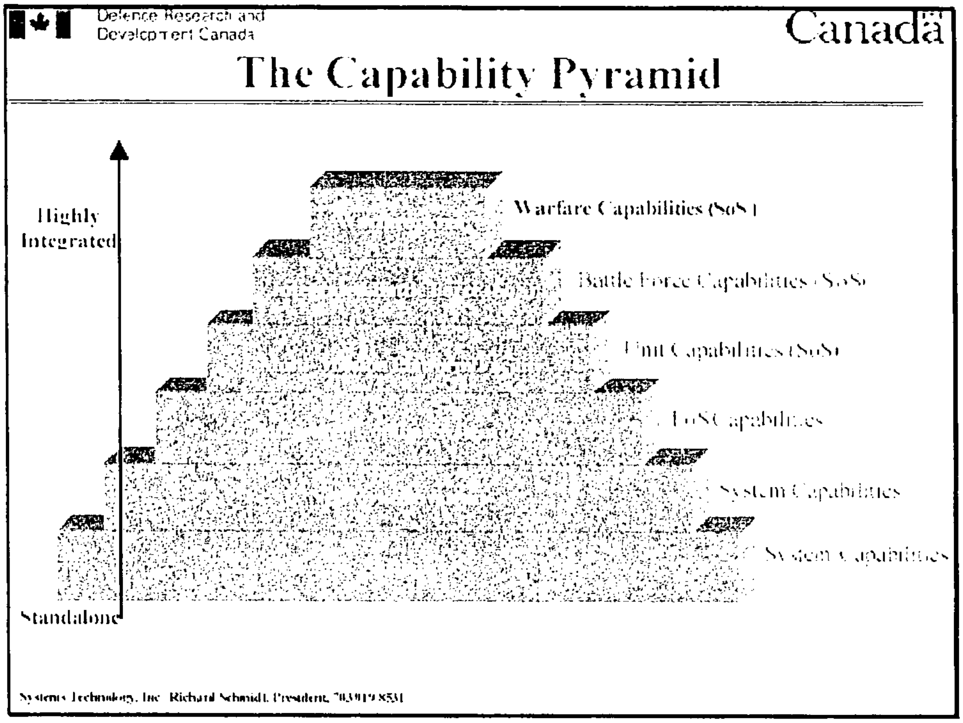

Figure 14: System Complexity Continuum [50, Slide 20]

Furthermore, all current and emerging standards still do not address the SysEng practice at a Capability Level. Figure 15 shows a generic SysEng Standard process, its current application to traditional systems, and its required application at a Capability level.

A separate section of this report covers in more detail the current status of SysEng standards.

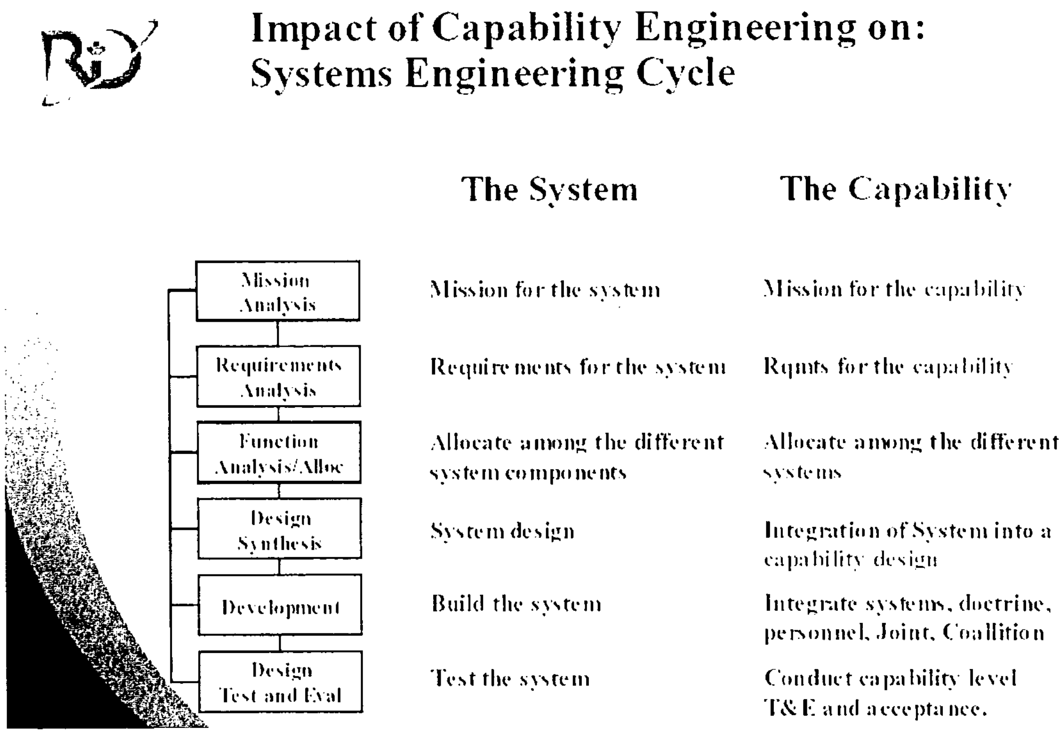

Figure 15: Impact of Capability Engineering on the Systems Engineering Cycle [49, Slide 21]

3.1.3 SoS vs. Single System

System of Systems (SoS) is a very important concept that applies to DND/CF systems. This special class of systems presents many challenges and issues of its own that we briefly introduce hereafter. An in-depth discussion of the SoS concept is provided as a separate document by Sweetnam and Harvey [40].

As described by Chen and Clothier [37], SoS evolution constitutes the main challenge for SysEng to be applied to a SoS context for the following reasons:

- ***System life cycle change:*** the life cycles of component systems merge into a life cycle of an SoS that may include a series of evolutions of its component systems over time. Thus, it differs from the traditional definition of the system life cycle in SysEng;

- ***Engineering "object" change:*** Throughout the SoS lifecycle, the "object" being engineered varies according to evolution scenarios. The "object" is often localized to a particular part of a SoS;
- ***Engineering focus change:*** Depending on the evolution scenarios, the engineering focus can be reflected in different evolution requirements from redesign, redevelopment, evolutionary development, integration, and new system components development; and
- ***Engineering environment change:*** Along with SoS evolution, different evolution scenarios have or need different engineering environments in terms of teams or stakeholders, information and knowledge resources, and supporting tools.

As discussed in greater detail in Sweetnam and Harvey [36], SysEng needs to address the challenges of a Systems of Systems defence environment. This environment is characterized by its independence of components, evolutionary nature, emergent behaviour and geographic extent. These factors, along with the problems and issues described in this report, will influence the appropriate selection and modification of SysEng processes, approaches and tools. As described in Chen and Clothier [37] SoS evolution is identified as the main challenge for applying SysEng to a SoS context. A number of existing SysEng concepts deal with system evolution issues, open systems approach, evolutionary development and product improvement strategies. The US DoD has also identified the following primary SysEng principles [41]:

- "Planning for system change;
- Applying the SysEng process;
- Managing interface changes;
- Identifying and using interface standards that facilitate continuing change;
- Ensuring that life cycle management is implemented;
- Monitoring the need for system modifications; and
- Ensuring that operations, support activities, and early field results are considered in planning."

However, as described by Chen and Clothier [37], an overall integration framework is required to provide a sufficient engineering solution to support the complexity of a SoS environment. An overall integration framework should address SysEng Organization, Infrastructure, Management, Interoperability and Architecture:

- ***Systems Engineering Organization:*** addresses the challenges of a highly complex and dynamic environment in applying SysEng practices at the level of individual system evolutions. Concurrent engineering can be applied to support these evolutions with attention given to the interrelations and dependency between systems, the relevancy between evolutions and the individual systems life cycle;
- ***Systems Engineering Infrastructure:*** is required to enable the SysEng activities to support SoS development. The elements of such an infrastructure would include:

    - Development of an integrated SysEng management plan;
    - Enhanced engineering process management and coordination;
    - Integrated standards guidance;
    - Integration with other disciplines such as software engineering, domain engineering, architecture practice, etc.;
    - Disciplined architecture practice; and
    - Determination of requirements and strategies supporting interoperability.

  In order to address these infrastructure elements an organization needs to define:

    - A joint SysEng framework to be used by all;
    - A SysEng structure;
    - A set of defined processes for all important engineering activities;
    - A set of SysEng standards;
    - A SysEng Information Management System (a repository of information/knowledge for all engineering artefacts) plus necessary tools supporting distributed and concurrent engineering, simulation and testing;
    - An architecture practice; and
    - A SysEng education and training program.
- ***Systems Engineering Management:*** is an approach that addresses the SoS challenges. It requires a modern SysEng practice that manages engineering activities and data/information across projects and systems domains. With a Systems Engineering Infrastructure in place, a set of procedures need to be defined that will clearly scope the engineering activities to be performed by a SysEng team within a SoS context. The key issues here are communication and collaboration. Chen and Clothier [37] propose a SysEng management paradigm that addresses these requirements.
- ***Interoperability:*** In order to examine and plan interoperability between systems, an Interoperability Maturity Model is required. DoD has developed the Levels of Information Systems Interoperability (LISI) – see Figure 16. This model provides a common basis for requirements definition and for incremental system improvements. The purpose of LISI is to provide DoD with a maturity model and a process for determining joint interoperability needs, assessing the ability of information systems to meet those needs, and selecting pragmatic solutions and a transition path for achieving higher states of capability and interoperability. LISI is a process for defining, evaluating, measuring, and assessing information systems interoperability. LISI uses a common frame of reference and measure of performance.

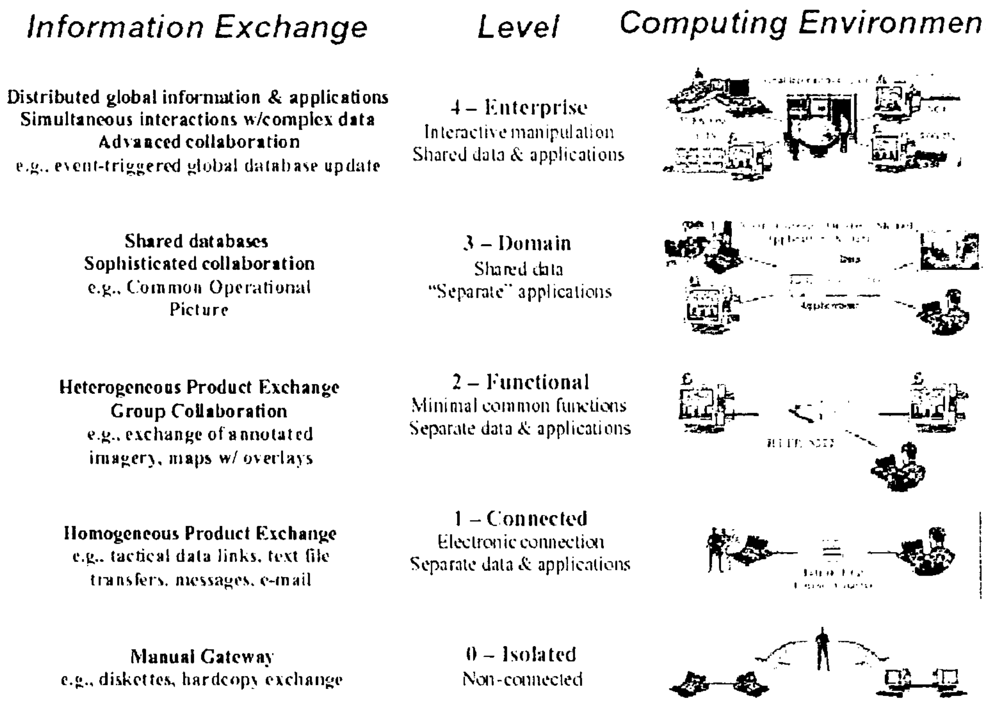

Figure 16: LISI levels of interoperability sophistication [42]

- ***Architecture:*** is a key element in addressing the complexities in a SoS. Therefore an architecture practice needs to be effectively planned, rationalized and combined with SysEng activities throughout the system life cycle management required for SoS. The DoDAF version 1 provides such a framework, however, a mapping of the architectural artefacts to the SysEng processes is not well defined and the DoDAF may also be lacking certain views to support the complete systems life cycle. Chen and Clothier [37] suggest the development of an architecture process framework that defines architecture activities and products required, and aligns them with the SysEng processes for SoS development and management. This framework can provide the architecture guidance to the System Engineering processes, and define the linkages between this framework and other architecture frameworks such as the DoDAF, Business Transformation Enablement Program (BTEP) and Zachman.

3.1.4 Long Acquisition Cycle

As observed by the Committee on the Future of the U.S. Aerospace Infrastructure and Aerospace Engineering Disciplines [41], long acquisition cycle times may impact the quality of the capabilities delivered to operational forces since systems may be based on old requirements and obsolete technologies. Moreover, this may in turn have a significant impact on the defence organizations and infrastructures that design, develop and implement these systems.

Long acquisition cycles are also major contributors to program instability, the most disruptive of which is the higher rate at which costs increase. Cost increases in one program generally mean that budgets elsewhere must be reduced or programs cancelled to make up the difference.

Cancellation is the ultimate form of instability. In a rapidly changing environment, programs based on ten- or twelve-year-old requirements become increasingly vulnerable to cancellation. Unfortunately, programs cancelled late in the development process result in zero return on very large investments. In addition to these huge losses, program cancellations may be a major contributor to the difficulty of retaining specifically trained, skilled, and experienced workers.

Changes in program direction may also occur as the result of changes in leadership. During a fifteen-year acquisition cycle, the leadership at each level may turn over several times. These changes are generally reflected in shifting priorities and associated budget cuts and redirection.

3.1.5 Changing Requirements

While the problem of poor requirements engineering and management has been repeatedly and widely discussed and documented as a contributing cause of project failures, major efforts are being made at DND/CF and abroad to enhance the requirements engineering process. However, many factors such as the changing nature of threats, the multiplicity of stakeholders, long acquisition cycles, and the dynamic nature of technology opportunities, drive the need for improvement notably in the areas of requirements tracking and impact assessment covering the whole capability life cycle.

3.1.6 Integrated and Dispersed Project Teams

The design of military systems (i.e. capabilities) involves optimizing over many competing disciplines and objectives, and requires a broad range of technical expertise that is not necessarily resident within a single organization. The optimization and development of these systems may thus require the collaboration of several government, academic, and commercial organizations.

Although computer-based collaboration addresses many of the management and coordination problems associated with paper-based systems, several technical and psychological barriers must still be crossed in order to achieve the expected level of efficiency. These barriers to collaboration fall into two general categories. First are the technical problems associated with integrating diverse, dispersed computational tools and hardware, with the added concerns and requirements for the security of classified information. The second type of barrier is cultural or psychological.

These barriers are carried over from the paper-based age, and in many cases are exacerbated by the increased physical separation of stakeholders, by the increased ability to move information, and by the lack of established processes to guide organizational interaction.

3.2 Specific Challenges

The following specific challenges have been identified by the Advisory Committee on Administrative Efficiency [39], and reported in the analysis of the current situation regarding the capability decision–making process [40].

3.2.1 Equipment Platform-centric Approach to Strategic Planning

The current strategic planning process appears to be focused on the production of an equipment acquisition plan largely driven by a "bottom-up" process that collates the various requirements of the three Environments (i.e., Army, Air and Navy). The shortfalls of this "bottom-up" approach are exacerbated by the fact that current planning retains a focus on specific equipment "platforms' rather than on "capabilities".

3.2.2 Approach to Requirements Definition

Despite the adoption of Capability Engineering, too many capital equipment projects and underlying requirements are still driven "bottom-up" rather than "top-down" and do not flow from coherent overall capability and business plans. Moreover, Defence's internal process for defining requirements takes too long, involves too many authorities and committees, occupies too much senior management time for little added value, and fails to distinguish between processes on the basis of risk and complexity. As said earlier, significant efforts and investments have been made by DND/CF to improve the Requirements Engineering process and environment. One of the major challenges lies in the areas of requirements tracking and impact assessment covering the whole capability life cycle.

3.2.3 Approach to Capability Acquisition

As reported by the Minister's Advisory Committee on Administrative Efficiency [39], the current approach to capability acquisition within the Canadian forces still needs to be improved despite the ongoing initiatives:

- "Low tolerance to risk throughout Defence is exemplified by "one size fits all" approach to capital expenditure approval process and by the organization's tendency to manage by committee;
- Defence's internal process for approving capital projects takes too long, involves too many authorities and committees, occupies too much senior management time for little added value, and fails to distinguish between processes on the basis of risk and complexity;
- Procurement – which includes the Acquisition process – is universally viewed as being a slow and cumbersome process that does not fully respond to Defence's needs;
- Acquisition of major military systems takes too long, with the average being over fifteen years for major capital equipment procurement;
- Substantial duplication of effort or functional overlap between DND and PWGSC;
- DND's internal approval process involves excessive non-value-added review and an undifferentiated approach to risk management;
- The total value of projects approved for inclusion in the LTCP far exceeds available funding, yet projects included in the plan with little likelihood of approval consume staff resources and administrative overhead; and
- Capital projects are not always closed in a timely fashion."

3.2.4 Formal Methods

Despite the improvements being pursued by the ongoing capability management initiative, the acquisition reform and others, there is still a well-entrenched perception that there is a lack of formal methods to evaluate and review projects, select options, and evaluate risks.

4. Recommendations for CapDEM TD

In order to address the many challenges in acquiring systems within the Defence community, CapDEM TD should further explore the recommendations described in the following sections.

4.1 Address the Specific Nature of Military Systems

The many unique differences that characterize military systems require that specific principles SysEng principles be applied at every stage of the acquisition cycle and to the supporting management and technical processes. These principles can be summarized as:

- Identification of all project stakeholders and capturing and reconciling of their requirements and expectations;
- Adoption of a "through life" approach to the project;
- Optimization of the whole system rather than component sub-systems;
- Placing focus on sub-systems integration when partitioning the system into sub-systems, and during testing and evaluation;
- Identifying all interfaces within the system, and between the system and its environment and other systems; and
- Remembering that everything is connected to everything else.

4.2 Focus on Capabilities

Implement a Capability Engineering process within an overarching Capability Management Framework in order to bring the rigour of SysEng and a robust analytical environment to support the acquisition of capabilities as defined through capability-based planning.

4.3 Address SoS Challenges

The selected SysEng approach needs to follow a holistic through-life approach that addresses the delivery of DND/CF capabilities within a Systems of Systems environment. Further detail provided in Sweetnam and Harvey [36].

Implement an overall systems integration framework that addresses SysEng Organization, Infrastructure, Management, Interoperability and Architecture:

- ***Systems Engineering Organization:*** Concurrent engineering should be applied to support a SoS environment with attention given to the interrelations and dependency between systems, the relevancy between evolutions and the individual systems life cycle.
- ***Systems Engineering Infrastructure:*** Implement a SysEng infrastructure that would provide:
    - Engineering data/knowledge sharing;

- Development of an integrated SysEng management plan;
- Enhanced engineering process management and coordination;
- Integrated standards guidance;
- Integration with other disciplines such as software engineering, domain engineering, architecture practice, etc.;
- Disciplined architecture practice; and
- Determination of requirements and strategies supporting interoperability.

In order to address these infrastructure elements CapDEM TD would need to define:

- A joint SysEng framework to be used by all stakeholders in the defence community;
- A SysEng structure;
- A set of defined processes for all important engineering activities;
- A set of SysEng standards;
- A SysEng Information Management System (a repository of information/knowledge for all engineering artefacts) plus necessary tools supporting distributed and concurrent engineering, simulation and testing;
- An architecture practice; and
- A SysEng education and training program.

- *Systems Engineering Management:* Define a SysEng management approach that implements a SysEng infrastructure. A set of procedures need to be defined that will clearly scope the engineering activities to be performed by a SysEng team within a SoS context. Chen and Clothier [37] propose a SysEng management paradigm that addresses these requirements. This proposed paradigm should be further explored.
- *Interoperability:* Adopt the DoD Interoperability Maturity Model Levels of Information Systems Interoperability (LISI) which will provide a common basis for determining joint interoperability needs, assessing the ability of information systems to meet those needs, and selecting pragmatic solutions and a transition path for achieving higher states of capability and interoperability.
- *Architecture:* Adapt the DoDAF version 1 to support the SysEng activities as shown in Figure 11. Additional research is required to identify SysEng processes that can benefit from an architectural description and determine the architectural framework that could satisfy those requirements if not addressed within the DoDAF such as BTEP [51]. Further explore Chen and Clothier [37] suggested architecture process framework that would define architecture activities and products, and align them with the SysEng processes for SoS development and management.

4.4 Adopt Evolutionary Systems Engineering

Long acquisition timescales and the changing environment which characterize military systems can lead to capabilities which, when put into service, are found to be obsolete or fail to meet user needs. This situation "militates" in favour of adopting an Evolutionary SysEng Approach that first delivers an initial operational capability (IOC) followed by incremental improvements. This approach brings successive

refinements that consider changing user requirements and take advantage of rapidly evolving technology opportunities.

The long acquisition cycle characterizing military systems is a major cause of changing user requirements, and the adoption of an Evolutionary Systems Engineering approach is without doubt the best one to adopt in the defence environment. However, this requires that we maintain a link between the changing users needs and the actual capability achieved throughout the entire system life cycle.

4.5 Adopt Collaborative Engineering

In order to remove the technical, cultural and psychological barriers associated with the design of military systems, it will be necessary to:

- Adopt and enforce the concepts of concurrent engineering;
- Define clear organizational responsibilities to warrant the good stewardship of the above environment and procedures.
- Define appropriate standards-based processes and procedures to guide the whole spectrum of through life SysEng activities;
- Acquire and use a repository-based collaborative engineering environment which includes all necessary functionality and connectivity to support the standard processes and the particular requirements of varied and dispersed user community; and
- Conduct adequate change management and training programs to ensure the adoption, appropriate use, and continuous evolution of the new environment.

4.6 Review Current Approaches

Many of the solutions to the inadequacy of current approaches to Strategic Planning, Requirements Definition and Capability Acquisition are not within the possibilities offered by the SysEng paradigm. For example, part of those solutions involves organizational and policy changes. A substantial portion of those solutions is currently in the process of being implemented as part of the ongoing acquisition reform and the front-end harmonization initiative being realized under the CapDEM TD umbrella. However, the SysEng standards and approaches to be adopted for the Capability Engineering Process will have to be extended or/and adapted to place the emphasis on capabilities and provide support to a top-down and through life approach to requirements definition and refinement.

4.7 Adopt Formal Methods

In order to address the lack of formal methods CapDEM TD should explore the development of a SysEng guide that will identify not only the standards and processes to be adapted but also the 'how-to' apply them. As a minimum CapDEM TD should:

- Adopt the ISO 15288, EIA 632, and IEEE 1220 as guiding standards for Systems Engineering;

- Select and tailor as necessary the applicable processes from the ISO 15288 system life cycle based on the system life cycles stages to be supported;
- Select and tailor as necessary the applicable processes from EIA 632 for the concept, development and production stages;
- Select and tailor as necessary the applicable processes from IEEE 1220 for the development stage;
- Determine any other engineering standards to be applied and select the applicable processes; and
- Implement Capability Maturity Model Integration for Systems and Software Engineering.

5. Conclusion

The capabilities-based methodology that DND wishes to develop is intended to facilitate force planning in an uncertain environment by identifying the broad set of capabilities required for the next strategic planning cycle (out to approximately 2020). This shift away from threat-based to capability-based force structure planning will require a capability-based engineering approach based upon a sound Systems Engineering methodology and SoS framework.

This report is the results of a 40-days effort summarizing the state of the art of the SysEng field. It is in no way exhaustive and reflects the most obvious findings and conclusions that could be obtained in this short period of time.

Systems Engineering best practices are embodied in the various standards, processes, methods, frameworks and approaches described in this report. Numerous challenges in the acquisition of defence capabilities have been identified that could be improved with adopting and tailoring a formal Systems Engineering methodology. Recommendations for CapDEM TD on the potential Systems Engineering practices to be explored that can address these challenges have been identified.

6. References

1. IEEE 610.12-1990 Standard Glossary for Software Engineering Terminology

2. IEEE 1220-1998 Standard for Application and Management of the Systems Engineering Process

3. ISO/IEC 15288-2002 Systems Engineering – System Life Cycle Processes

4. IEEE 1220-1994 Standard for Application and Management of the Systems Engineering Process

5. ISO/IEC 12207 Information Technology – Software Life Cycle Processes

6. EIA 632 Standard – Processes for Engineering a System

7. EIA 649 Standard for Configuration Management

8. International Council on Systems Engineering (INCOSE) website, http://www.incose.org/whatis.html

9. IDEAL: User Guide for Software Process Improvement, 1996

10. ISO/IEC 19760-2003 Systems Engineering - A Guide for the Application of ISO/IEC 15288 (System Life Cycle Processes)

11. NASA Systems Engineering Handbook, http://acc.dau.mil/simplify/ev.php?ID=7176_201&ID2=DO_TOPIC

12. Software Productivity Consortium, *"Integrated Systems and Software Engineering Process (ISSEP)"*

13. Boehm, Barry, *"Spiral Development: Experience, Principles and Refinement, Spiral Development Workshop"*, July 2000.

14. Carnegie Mellon – Software Engineering Institute (SEI), *"Capability Maturity Model® Integration (CMMI)"*, Version 1.1, March 2002

15. Carnegie Mellon – Software Engineering Institute (SEI), *"Capability Maturity Models®"*, http://www.sei.cmu.edu/cmm/cmms/cmms.html

16. Carnegie Mellon – Software Engineering Institute (SEI) http://www.sei.cmu.edu/cmmi/general

17. Project Management Institute (PMI), *"Project Management Book of Knowledge"*, http://www.pmi.org/info/default.asp

18. Treasury Board Secretariat (TBS), *"Government of Canada, Business Transformation Enablement Program (BTEP)"*, http://www.cio-dpi.gc.ca/btep-phto/documents/2003/synopsis/synopsis_e.asp

19. Object Management Group (OMG), *"XML Metadata Interchange (XMI)"*, http://www.omg.org/technology/documents/formal/xmi.htm

20. Object Management Group (OMG), *"Unified Modelling Language"*, UML Resource Page, http://www.uml.org/

21. Systems Modelling Language (SysML), http://www.sysml.org/

22. Defence Acquisition University (DEA), Acquisition Community Center, *"NASA Systems Engineering Handbook"*, http://acc.dau.mil/simplify/ev.php?ID=7176_201&ID2=DO_TOPIC

23. Defence Engineering Group (DEG), University College London (UCL); *"The Defence Systems Engineering Handbook"*- Written for the Ministry of Defence (UK), 2002; http://www.deg.meng.ucl.ac.uk/DSE_GUIDE_EDITION_1A.PDF

24. Parth, Frank, R., *"Systems Engineering Drivers in Defence and in Commercial Practice"*, Deloitte & Touche Consulting Group, Feb 15, 1998

25. Lamartin, Glen, F., Director Defence Systems DoD, *"Systems Engineering – Keynote Address"*, January 26, 2004

26. Ministry of Defence (MOD), United Kingdom (UK), *"Acquisition Management System"*, http://www.ams.mod.uk

27. Ministry of Defence (MOD), United Kingdom (UK): *"Acquisition Policy and Process Framework"*, http://www.ams.mod.uk/ams/default.htm

28. International Council on Systems Engineering (INCOSE), *"Systems Engineering Handbook"*, Version 2.0, July 2000

29. Software Productivity Consortium, *"Integrated Systems and Software Engineering Processes"*, Version 01.00.04, May 1996, http://www.software.org

30. CERN Engineering Data Glossary, http://cedar.web.cern.ch/CEDAR/glossary.html

31. Open Systems Joint Task Force, *"An Open Systems Approach to Weapon System Acquisition"*, Working Draft, Version 1.0, http://www.acq.osd.mil/osjtf/msword/OS%20Guide%20Section%201.doc

32. Wilson, M. and Mooz, H., *"Agile Systems Engineering for Rapid Product Development"*, Center for Systems Management, INCOSE, 2003.

33. Dickerson, C., E., Soules, S. M., Sabins, M. R., Charles, P. H., *"Using Architectures for Research, Development and Acquisition"*

34. Object Management Group, *"Model-Driven Architecture"*, http://www.omg.org/mda/

35. Drouin, C., Wellwood, M., *"CapDEM TD - "Modelling and Simulation (Role), State of the Art Report"*, DRDC Valcartier, 2004

36. Sweetnam, J., P., Harvey, G., *"System of Systems - State of the Art Report"*, DRDC Valcartier, 2004

37. Chen, P., Clothier, J., *"Advancing Systems Engineering for Systems of Systems Challenges"*, Systems Engineering, Vol 6, No 3, 2003

38. Committee on the Future of the U.S. Aerospace Infrastructure and Aerospace Engineering Disciplines to Meet the Needs of the Air Force and the Department of Defense, *"Review of the Future of the U.S. Aerospace Infrastructure and Aerospace Engineering Disciplines to Meet the Needs of the Air Force and the Department of Defense"*, National Research Council, National Academy Press, Washington, D.C., 2001, http://www.nap.edu/catalog/10201.html

39. Minister's Advisory Committee on Administrative Efficiency, *"Report to the Minister of National Defence by the Advisory Committee on Administrative Efficiency"*, Aug 21, 2003

40. Harvey, G., Leclerc, J., *"DND Capability Decisision-Making Process: The 'As-is'"*, DRDC Valcartier, Mar 24, 2004

41. US Department of Defense (DoD) – Systems Management College Press, *"Systems Engineering Fundamentals"*, January 2001

42. DoD Architecture Framework Working Group, *"DoD Architecture Framework Deskbook"*, Version 1, August 2003, http://www.aitcnet.org/dodfw/Deskbook.zip

43. DoD Adoption Notice, http://www.acq.osd.mil/io/se/Adoption%20Notice%20for%2015288.doc

44. DRDC Valcartier, CapDEM TD Fact Sheet IS-224-A, October 2003

45. Bagnulo A., Adapted from Software Productivity Consortium's Frameworks Quagmire, http://www.software.org/quagmire

46. ISO/IEC 15504 International Standard for Software Process Assessment, http://www.sei.cmu.edu/cmmi/faq/15504-faq.html#Q241

47. EIA/IS 731 Interim Standard for Systems Engineering Capability Model, http://www.sei.cmu.edu/cmmi/faq/map-faq.html#MAP03

48. Boehm, Barry, USC, CSE and Hansen W.J, CMU, SEI, *"Understanding the Spiral Model as a Tool for Evolutionary Acquisition"*, January 2001, http://sunset.usc.edu/publications/TECHRPTS/2001/usccse2001-501/usccse2001-501.pdf

49. Bovenkamp J. and Pagotto J., DRDC Ottawa, *"Synthetic Environment Integrated with Capability Engineering – Powerful Tools to Address the Future", Slide 17*, Presentation, 15 April 2003

50. Crisp Dr. H., RDA CHEng, *"Battle Force Systems Engineering Process"*, Slide 20, Presentation to RDDC Ottawa, 7 February 2002

51. Treasury Board of Canada Secretariat (TBS), *"Business Transformation Enablement Program", Version 1.5b, 18 February 2004,* http://www.cio-dpi.gc.ca/btep-phto/documents/2003/synopsis/synopsis_e.asp

52. Dickerson Dr. C.E., RDA CHEng, *"CIAP – Architecture Description"*, Presented at NDIA Systems Engineering Conference, Oct 21-24, 2002, http://www.dtic.mil/ndia/2002systems/dickerson2a5.pdf

List of Acronyms

BTEP	Business Transformation Enablement Program
CapDEM TD	Collaborative Capability Definition Engineering and Management Technology Demonstrator
CE	Capability Engineering
CEP	Capability Engineering Process
CF	Canadian Forces
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
COI	Capacité opérationnelle initiale (IOC French equivalent)
COT	Capacité opérationnelle totale (FOC French equivalent)
DIGCap	Définition, Ingénierie et Gestion Collaboratives des Capacités (CapDEM French equivalent)
DND	Department of National Defence
DoD	US Department of Defence
DoDAF	US Department of Defence Architecture Framework
DSE	Defence Systems Group
DT	Démonstration de technologie (TD French equivalent)
EIA	Electronics Industries Association
ÉU	États-Unis
FC	Forces canadiennes
FOC	Final Operational Capability
IEC	International Electrotechnical Commission

IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council On Systems Engineering
IngCap	Ingénierie des capacités
IngSys	Ingénierie de systèmes (SysEng French equivalent)
IOC	Initial Operational Capability
ISO	International Organization for Standardization
ISSEP	Integrated Systems and Software Engineering Process
LISI	Levels of Information Systems Interoperability
LTCP	Long-Term Capital Plan
M&S	Modelling and Simulation
MDN	Ministère de la Défense nationale
MOE	Measures of Effectiveness
NASA	National Aeronautics and Space Administration
PIC	Processus d'ingénierie des capacités (CEP French equivalent)
PMBok	Project Management Book of Knowledge
PWGSC	Public Works and Government Services Canada
RU	Royaume-Uni (UK French equivalent)
SBA	Simulation-based Acquisition
SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SdS	Système(s) de systèmes
SoS	System(s) of Systems
SysEng	Systems Engineering
SysML	Systems Modelling Language
TD	Technology Demonstration

UAV	Unmanned Airborne Vehicle
UK MoD	United Kingdom Ministry of Defence
UML	Unified Modelling Language
XMI	XML Metadata Interchange
XML	Extensible Markup Language

Distribution List

INTERNAL DISTRIBUTION

2 - Document Library

1 - F. Bernier

1 - M. Couture

1 - G. Dussault

1 - C. Lalancette

1 - M. Lizotte

1 - M. Mokhtari

EXTERNAL DISTRIBUTION

1 - DRDKIM

SANS CLASSIFICATION

COTE DE SÉCURITÉ DE LA FORMULE
(plus haut niveau du titre, du résumé ou des mots-clefs)

FICHE DE CONTRÔLE DU DOCUMENT

1. PROVENANCE (le nom et l'adresse) CGI	2. COTE DE SÉCURITÉ (y compris les notices d'avertissement, s'il y a lieu) Sans classification

3. TITRE (Indiquer la cote de sécurité au moyen de l'abréviation (S, C, R ou U) mise entre parenthèses, immédiatement après le titre.)
CapDEM TD - Systems Engineering State of the Art Report

4. AUTEURS (Nom de famille, prénom et initiales. Indiquer les grades militaires, ex.: Bleau, Maj. Louis E.)
Attilio Bagnulo (CGI) Jocelyn Leclerc (CGI)
Authorité Scientifique - Mario Couture

5. DATE DE PUBLICATION DU DOCUMENT (mois et année) 2005	6a. NOMBRE DE PAGES 67	6b. NOMBRE DE REFERENCES 52

7. DESCRIPTION DU DOCUMENT (La catégorie du document, par exemple rapport, note technique ou mémorandum. Indiquer les dates lorsque le rapport couvre une période définie.)
Rapport de contrat

8. PARRAIN (le nom et l'adresse)

9a. NUMÉRO DU PROJET OU DE LA SUBVENTION (Spécifier si c'est un projet ou une subvention) 15as (Numéro de projet)	9b. NUMÉRO DE CONTRAT W7701-3-2621
10a. NUMÉRO DU DOCUMENT DE L'ORGANISME EXPÉDITEUR	10b. AUTRES NUMÉROS DU DOCUMENT N/A

11. ACCÈS AU DOCUMENT (Toutes les restrictions concernant une diffusion plus ample du document, autres que celles inhérentes à la cote de sécurité.)

- [x] Diffusion illimitée
- [ ] Diffusion limitée aux entrepreneurs des pays suivants (spécifier)
- [ ] Diffusion limitée aux entrepreneurs canadiens (avec une justification)
- [ ] Diffusion limitée aux organismes gouvernementaux (avec une justification)
- [ ] Diffusion limitée aux ministères de la Défense
- [ ] Autres

12. ANNONCE DU DOCUMENT (Toutes les restrictions à l'annonce bibliographique de ce document. Cela correspond, en principe, aux données d'accès au document (11). Lorsqu'une diffusion supplémentaire (à d'autres organismes que ceux précisés à la case 11) est possible, on pourra élargir le cercle de diffusion de l'annonce.)

13. SOMMAIRE (Un résumé clair et concis du document. Les renseignements peuvent aussi figurer ailleurs dans le document. Il est souhaitable que le sommaire des documents classifiés soit non classifié. Il faut inscrire au commencement de chaque paragraphe du sommaire la cote de sécurité applicable aux renseignements qui s'y trouvent, à moins que le document lui-même soit non classifié. Se servir des lettres suivantes: (S), (C), (R) ou (U). Il n'est pas nécessaire de fournir ici des sommaires dans les deux langues officielles à moins que le document soit bilingue.)

Ce rapport fournit une revue de l'état de l'art concernant les différentes normes et approches d'ingénierie de systèmes pouvant s'appliquer à la nature complexes des systèmes de systèmes militaires. Il contient une analyse des pratiques et tendances actuelles concernant les standards et modèles de maturité, un survol des manuels et guides ainsi qu'une description des approches les plus utilisées. Le rapport discute également des problèmes et enjeux génériques associés aux systèmes militaires incluant un rappel des problèmes et enjeux qui sont spécifiques au contexte MDN/FC. Finalement le rapport énonce un ensemble de recommandations pouvant s'appliquer au contexte de DIGCap DT, à la lumière des connaissances acquises au cours de la revue de l'état de l'art.

14. MOTS-CLÉS, DESCRIPTEURS OU RENSEIGNEMENTS SPÉCIAUX (Expressions ou mots significatifs du point de vue technique, qui caractérisent un document et peuvent aider à le cataloguer. Il faut choisir des termes qui n'exigent pas de cote de sécurité. Des renseignements tels que le modèle de l'équipement, la marque de fabrique, le nom de code du projet militaire, la situation géographique, peuvent servir de mots-clés. Si possible, on doit choisir des mots-clés d'un thésaurus, par exemple le "Thesaurus of Engineering and Scientific Terms (TESTS)". Nommer ce thésaurus. Si l'on ne peut pas trouver de termes non classifiés, il faut indiquer la classification de chaque terme comme on le fait avec le titre.)

Acquisition Process

Capability

Capability Engineering

Capability Engineering Process

CapDEM

System Engineering

#523366
CA025673